Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université de Sousse
Institut Supérieur d'Informatique et des Technologies de Communication - Sousse
المعهد العالي للإعلامية و تكنولوجيات الاتصال بسوسة

Recueil des publications scientifiques

Présenté en vue de l'obtention de l'
Habillitation Universitaire

Spécialité
Informatique

Par
Khaled Jouini
Docteur de l'université Paris Dauphine-PSL (France)
Maître Assistant à l'ISITCom

Janvier 2025

.

Table des publications

.

Publications dans des revues internationales

.

Q1.1 Connecting the Dots between Stance and Fake News Detection with Blockchain, Proof of Reputation, and the Hoeffding Bound

Ilhem Salah, Khaled Jouini, Cyril-Alexandre Pachon & Ouajdi Korbaa

Cluster Computing. 2024.
ISSN : 1386-7857, Springer Nature.
JCR IF : 3.6.

DOI : https://doi.org/10.1007/s10586-024-04637-7
SJR best quartile : Q1, SJR : 1.07

Cluster Computing

COUNTRY

Netherlands

Universities and research institutions in Netherlands

Media Ranking in Netherlands

SUBJECT AREA AND CATEGORY

Computer Science
Computer Networks and Communications
Software

PUBLISHER

Springer Nature

H-INDEX

69

PUBLICATION TYPE

Journals

ISSN

13867857

COVERAGE

1998, 2005-2023

INFORMATION

Homepage

How to publish in this journal

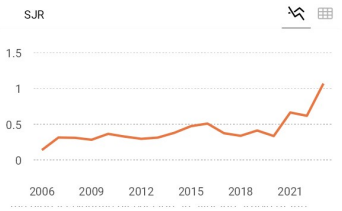
SCOPE

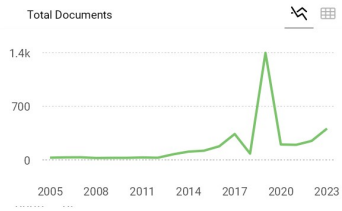
Cluster Computing: the Journal of Networks, Software Tools and Applications will provide a forum for presenting the latest research and technology that unify the fields of parallel processing, distributed computing systems and high performance computer networks. The current advances in computing, networking technology and software have spurred a lot of research interest in cluster and internet computing, as demonstrated in Cloud and Grid computing, and distributed high performance data centers. In the last few years, we have seen an increased interest in developing applications, software tools, communications protocols and high performance data centers, Grids and cloud computing sites to capitalize on these advances and initiatives. Publications about these developments currently appear in several journals that either focus on the communications field, or on parallel and distributed computing with a strong emphasis on the parallel algorithms. Cluster Computing Journal will uniquely address the latest results in integrating these three fields to support the development of high performance parallel distributed computing systems and their applications. The journal will be an important source of information for the growing number of researchers, developers and users of High Performance Parallel and Distributed Computing environments. In these environments, parallel and/or distributed computing techniques are applied to the solution of large-scale scientific and engineering applications running on clusters, cloud computing and/or distributed data centers.

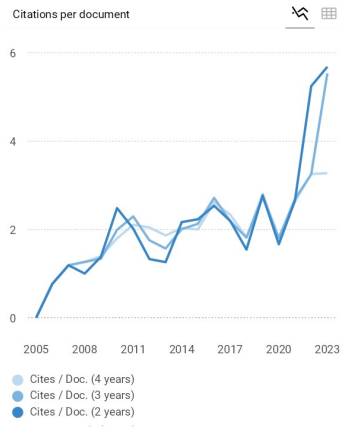
FIND SIMILAR JOURNALS

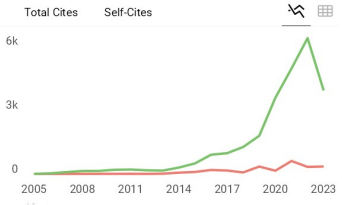
options

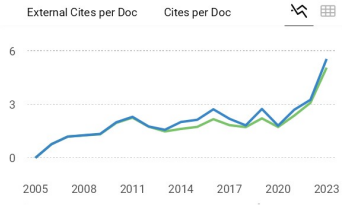
1	2	3	4	5
Journal of Cloud Computing	**Future Generation Computer Systems**	**Computing (Vienna/New York)**	**Concurrency Computation Practice and Experience**	**Journal of Grid Computing**
USA	NLD	AUT	GBR	NLD
66% similarity	63% similarity	61% similarity	61% similarity	60% similarity

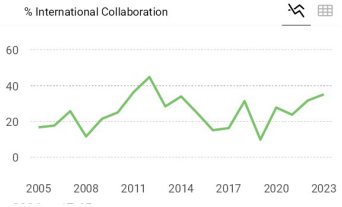


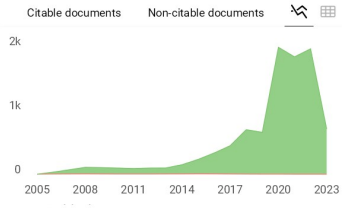


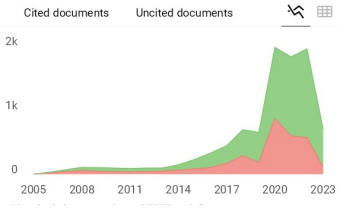


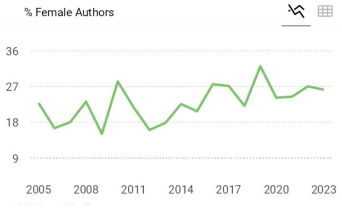


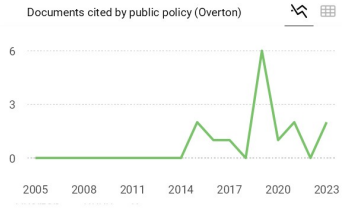


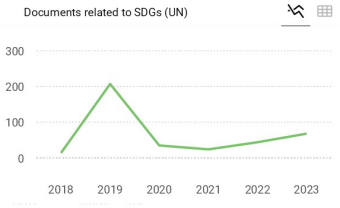












← Show this widget in your own website

Just copy the code below and paste within your html code:

<a href="https://www.scima

SCImago Graphica

Explore, visually communicate and make sense of data with our **new data visualization tool**.

Metrics based on Scopus® data as of March 2024

Cluster Computing
https://doi.org/10.1007/s10586-024-04637-7

Connecting the dots between stance and fake news detection with blockchain, proof of reputation, and the Hoeffding bound

Ilhem Salah[1] · Khaled Jouini[1] · Cyril-Alexandre Pachon[2] · Ouajdi Korbaa[1]

Received: 12 February 2024 / Revised: 11 April 2024 / Accepted: 13 June 2024

Abstract
Combating fake news is a crucial endeavor, yet the complexity of the task requires multifaceted approaches that transcend singular technological solutions. Traditional fact-checking, often centralized and human-dependent, faces scalability and bias challenges. This paper introduces a novel blockchain-based framework that leverages the wisdom of the crowd for an authority-free, scalable, automated and reputation-driven fact-checking. Within this framework, stance detection acts as an automated means of opinion retrieval, while the Proof of Reputation consensus mechanism fosters an environment where reputable contributors have greater influence in shaping news credibility. Concurrently, the Hoeffding bound is used to allow the system to adapt to evolving contexts. In contrast to Machine Learning—based approaches, our framework limits the need for periodic retraining to update a model's frozen knowledge of the world. The experimental study conducted on real-world data demonstrates that the proposed framework offers a promising and efficient solution to combat the spread of fake news.

Keywords Stance & Fake news detection · Blockchain technology · Proof of reputation · Hoeffding bound · Decentralized fact-checking

1 Introduction

In the era of ubiquitous Internet and social media platforms, where information of diverse types is readily available and where any point of view can find an audience, access to information is no longer an issue. The key challenges, instead, lie in veracity, credibility, and authenticity [1, 2]. Assessing the veracity of a news story is however a very complex and cumbersome task [3, 4], not only from a machine learning and Natural Language Processing (NLP) perspective, but also sometimes for the most experienced journalists and trained experts [3, 5]. That is why the scientific community often approaches the task from a variety of angles, moving beyond singular technological solutions and breaking down the process into independent sub-tasks. One practical step towards automated fact-checking involves estimating the opinion or point of view (*i.e., stance*) of different news sources regarding the same topic or claim [3]. This work explores the integration of *stance detection* as a key component of a decentralized, fully automated, end-to-end fact-checking pipeline built upon blockchain technology. Our ultimate objective is to derive a *credibility score* for claims based on the stances taken by various sources, with the stances being *weighted* by the respective sources' *reputation*.

Conventional fact-checking systems, often reliant on manual and centralized validation processes (i.e. central authority), are inherently constrained by scalability issues, potential biases, and censorship risks. Similarly, while valuable, conventional machine learning-based approaches often require periodic retraining to update their static and

✉ Khaled Jouini
j.khaled@gmail.com

Ilhem Salah
ilhemsalah53@gmail.com

Cyril-Alexandre Pachon
cyril.pachon@ecole-hexagone.com

Ouajdi Korbaa
ouajdi.korbaa@centraliens-lille.org

[1] MARS Research Lab LR17ES05, ISITCom, University of Sousse, 4011 Hammam-Sousse, Sousse, Tunisia

[2] École Hexagone, Parvis du Colonel Arnaud Beltrame, 78000 Versailles, France

Published online: 27 June 2024

potentially outdated knowledge of the world (as for example Large Language Models do). This necessity arises from the dynamic nature and continuous emergence of news events, posing challenges in scenarios where timely detection of fake news is crucial to prevent its spread. Additionally, conventional ML models heavily rely on manually annotated training data, which can be challenging to obtain in both quantity and quality, especially for specialized topics and live news events. Conventional ML-based approaches also often assume a centralized validation process and hence share the same limitations of conventional fact-checking regarding scalability issues and/or potential bias.

In recent years, blockchain technology, with its scalability and authority-free nature [6], is being harnessed to combat the spread of fake news [7]. Existing blockchain-based approaches have primarily focused on tracking the provenance and modification history of news articles and associated images [8–11] or on assigning veracity labels to them [12–14]. Blockchain-based veracity labeling approaches typically incorporate voting mechanisms and/or machine learning models to assess the credibility of news. While these efforts have yielded valuable insights and advancements, to the best of our knowledge, no prior research has employed blockchain technology to bridge the gap between stance detection and fake news identification, nor provided a statistical guarantee to control the risk that the label associated to a news changes over time when additional evidence (*e.g.*, votes or stances) becomes available.

In order to address the limitations of conventional fact-checking and machine learning-based approaches, this work introduces a novel decentralized framework built upon blockchain technology. By decentralizing the fact-checking process and harnessing the wisdom of the crowd, we aim to mitigate biases, improve scalability, and enhance the robustness of fake news detection. Within this framework, the credibility score of a news item is derived from the stances of participants engaging with it. To improve scalability and reduce user input, stance detection is used as an automated means to infer the opinions of participants towards a claim. A salient feature of our approach is the nuanced weighting (*i.e.*, reputation) assigned to each participant. This weight is determined according to their past positive interactions (*i.e.* supporting/sharing true news or refuting false news) and negative interactions (*i.e.* supporting/sharing false news or refuting true news). To safeguard the system against potential malicious actors and ensure trustworthy news assessment, our framework adapts the "Proof of Reputation" consensus mechanism to fake news detection by only allowing contributors with positive reputation to participate in the consensus for assessing news veracity. Stances typically arrive at irregular time intervals and waiting for all stances can delay fake news detection. To enable timely detection, our framework uses the Hoeffding bound as a statistical guarantee for labeling news items even with incomplete information.

The remainder of this paper is organized as follows. Section 2 briefly reviews the main existing blockchain-based fake news detection approaches. Section 3 presents our reputation-driven stance-based fake news detection framework. Section 4 presents an experimental study validating our approach. Finally, Sect. 5 concludes the paper.

2 Related work

Early efforts to leverage blockchain technology for fake news detection primarily focused on tracing information provenance and incentivizing the creation of reliable content. Noteworthy systems in this category include BlockProof [10], AnsaCheck [8], and the NY Times provenance project [9]. Inspired by several studies that have demonstrated the effectiveness of collective wisdom in surpassing the biases and limitations of individual experts [13, 15–17], more recent approaches have shifted towards the integration of crowdsourcing, voting, and machine learning models [18]. Our research aligns with the latter category of approaches. In this sequel, we mainly focus on two recent approaches, [12, 14], that effectively combine crowdsourcing and blockchain technology to combat fake news.

The study [12] proposes a decentralized social network using blockchain technology to manage news and an entropy-based incentive mechanism to encourage users to contribute to the system. The system uses a group of trusted human appraisers to evaluate the authenticity of news. The appraisers are chosen through an entropy-based algorithm that selects appraisers whose votes are likely to be more informative and less predictable. If a majority of appraisers vote that a news is fake, the news is flagged as such and its visibility is reduced. Appraisers are rewarded with tokens for flagging suspicious content, providing feedback on the accuracy of news articles, and participating in the consensus process. Token rewards are contingent upon the entropy of their actions. Users have the ability to rate appraisers based on the accuracy and helpfulness of their feedback. Appraiser reliability is assessed using various factors, including the percentage of correctly identified fake or real news articles, variance in votes for different news items, and reputation measured through feedback from other users. The primary concern of the approach of [12] is the reliance on manual voting by pre-selected appraisers, a method conflicting with natural interactions in social networks, and raising concerns regarding potential bias and scalability limitations. Furthermore, the effectiveness of the methodology depends on expert validators,

a form of central authority contradicting blockchain's decentralized spirit.

The approach proposed in [14] introduces a decentralized environment where users post and vote on news stories, utilizing the benefits of collective intelligence for information validation. The proposed approach incorporates Lyapunov exponents and Shannon entropy as key metrics to ascertain the equilibrium of votes on a given story and to decide when to end the voting procedure. Lyapunov exponents are employed to provide a quantitative measure of the degree of chaos in a system while Shannon entropy is used to measure the disorder of votes. Upon reaching equilibrium, the status (true or false) of a news story is determined through the utilization of machine learning classifiers. The system employs a two-stage learning architecture, comprising an Action Classifier and a Story Classifier. The Action Classifier, implemented as a convolutional neural network, is designed to discern malicious behavior, with a particular focus on identifying the involvement of bots. The Story Classifier utilizes Long Short-Term Memory (LSTM) networks to predict the label of a story based on the collective behavior of users who have voted on it. While presenting several interesting ideas, the approach of [14] exhibits two notable shortcomings. Firstly, the reliance on black-box machine learning models might pose challenges for users in understanding how decisions are made. This lack of transparency could raise concerns about the system's fairness and impartiality, crucial for gaining and maintaining user trust critical in tasks involving user-generated content. Secondly, the approach lacks a statistical guarantee when determining the veracity of a claim. This is particularly problematic since in the proposed approach the labels associated with news are immutable. Consequently, if credible votes later confirm a news story to be true after initially being labeled false, there is no recourse to alter its status. This rigidity coupled with the lack of statistical guarantee can be a limitation when dealing with evolving contexts.

It is worth noting that, to the best of our knowledge, our framework stands as the sole blockchain-based approach that avoids manual voting, provides a statistical guarantee when determining news veracity, and features dynamic and evolving reputation and credibility scores.

3 News credibility scoring with stances and reputations: a blockchain-based framework

3.1 Framework overview

In this paper, we present a novel framework that leverages stance detection and blockchain technology to establish an autonomous and decentralized system allowing to assign credibility scores to shared news and reputation scores to participants. To address the current limitations of blockchain platforms, we assume that news texts and machine learning models are stored off-chain and only participants' stances regarding news are stored on-chain. Our framework can be seen as a generalization of voting systems, wherein votes are inferred from stances rather than being explicitly provided by users. As highlighted by various studies, including [19], sharing and commenting news are the prevalent forms of news engagement among social media users and are more participatory and intuitive than voting and other paralinguistic digital affordances, which are more passive and impersonal actions.

The main steps of the news veracity assessment process are outlined in the flowchart of Fig. 1. As depicted in Fig. 1, participants interact with the system by commenting or sharing news items (Step 1). The system, in turn, detects the participant's stance regarding the news and ensures that the stance is either "Support" or "Deny" (Step 2). If not, the process concludes; otherwise, the stance is logged in

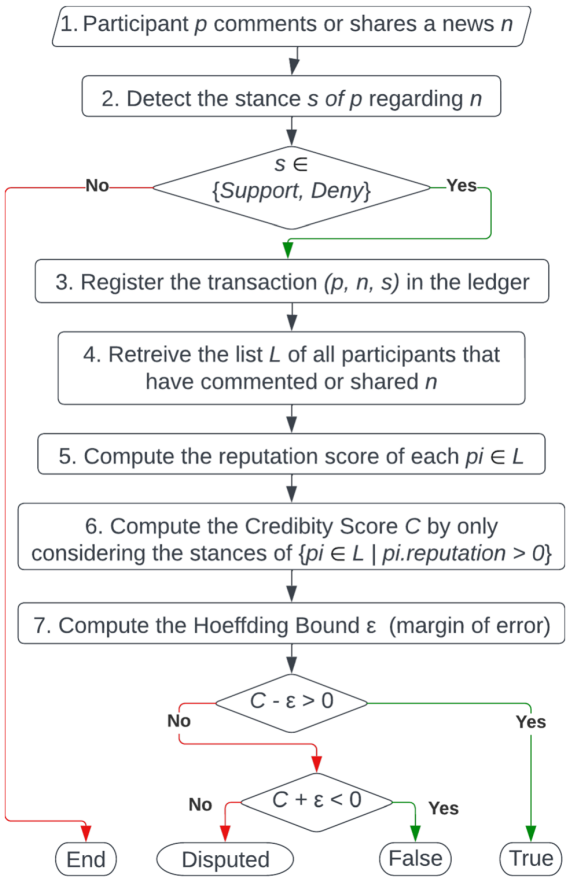


Fig. 1 Comment handling process

the distributed ledger ("Neutral" and "Unrelated" stances are not registered) (Step 3). Serving as a platform for tracking stance history, the distributed ledger facilitates the determination of both positive contributions and negative contributions made by each participant. The distributed ledger ensures that the history of contributions cannot be tampered with, thereby enhancing the overall integrity of the system. The smart contract associated with the distributed ledger enables the determination of participants' reputation and news credibility scores on the fly.

The credibility score of a news item depends on the stances and reputation scores of participants who commented or shared it. Reputation scores draw inspiration from conventional cryptocurrency balances, dynamically increasing with positive contributions and decreasing for negative actions. These "balances" reflect the overall behavior of participants within the system. The reputation score of a participant is computed by scanning their entire stance history (Step 5). This involves summing up positive contributions (similar to incoming transactions in a standard blockchain), deducting negative contributions (akin to outgoing transactions), and subsequently normalizing the result. The reputation score reflects therefore the trustworthiness of participants based on their interactions within the system.

Reputation acts as a trust indicator, guiding the consensus on news veracity assessments (Step 6). The consensus mechanism adopted in our framework bears similarities with Proof of Reputation (PoR) and Proof of Stake (PoS), akin to how these consensus mechanisms verify transactions in conventional blockchain. As shown in Fig. 1, only participants with positive reputations are allowed to contribute to the consensus, *i.e.*, only the stances of participants with positive reputations are taken into account when determining the credibility score of a news. This is analogous to how stakeholders in Proof of Stake systems exert more influence in blockchain consensus. Analogous to reputation scores, news credibility scores are dynamically recalculated as new stances become available. The credibility score C of a news item n is computed as the sum of the stances of participants who have a positive reputation score, where each stance is weighted according to the corresponding participant's reputation score.

After calculating a credibility score, the next step involves determining the status of the news story (that is, whether the news is "True," "False" or in a "Disputed" state) (Step 7). Accurate status determination is crucial for maintaining result consistency, as associating an incorrect label with news directly impacts the reputation scores of participants, potentially leading to adverse consequences for the entire system. While status determination is straightforward when all stances regarding a news item are available, in real-world scenarios and dynamic contexts, stances often arrive at irregular time intervals, and decisions need to be made dynamically before all stances become available. Using fixed credibility thresholds for status determination introduces various drawbacks, including over-generalization, limited adaptability to evolving data, and potential biases. In response to these challenges, our approach uses the Hoeffding Bound [20] to derive an adaptive margin of error ϵ with a statistical guarantee. As shown in the flowchart of Fig. 1, our framework evaluates if the credibility score, considering ϵ, aligns with labeling the news as "True," "False" or "Disputed."

In the remainder of this section, we first provide a detailed explanation of how the reputation and credibility scores are computed. Subsequently, we elaborate on how the Hoeffding Bound is used for determining news status.

3.2 On-demand assessment of participant reputation and news credibility

Participants' understanding of the reputation and credibility scores is crucial for ensuring transparency and encouraging active participation. Consequently, we have opted for highly straightforward formulas that make the concepts of reputation and credibility intuitive and easily understandable. By presenting these simplified formulas, participants can gain a clear understanding of how their actions directly influence their scores and contribute to the system's assessment of news article credibility.

3.2.1 Reputation score

As mentioned earlier, in our work we do not explicitly store participant reputation and news credibility scores. Instead, we employ an on-demand, real-time reputation balance approach, akin to traditional blockchain systems. Similar to a standard blockchain's ledger that maintains records of all transactions, we keep a record for each participant interaction within the network. Each interaction record includes the participant's address, news identifier, and the participant's stance regarding a news. The reputation score of a participant is determined based on the history of their stances regarding news items identified as either true or false. Since reputation is linked to past behavior and contributions, participants have a vested interest in maintaining a positive reputation. Additionally, the reputation score makes it difficult for malicious actors to manipulate the network by creating multiple identities (*i.e., Sybil attacks*), as their influence is constrained by cumulative reputation scores.

Algorithm 1 Reputation score calculation

```
function GETREPUTATION(particiantHash)
    for each news in participants[participantHash].commentedAndSharedNews do
        status ← newsMap[newsID].status
        stance ← participants[participantHash].stances[newsID]
        if (status × stance > 0) then
            positiveContributions++
        else if (status ≠ 0) then
            negativeContributions + +
        end if
    end for
    totalContributions ← positiveContributions + negativeContributions
    if (totalContributions) > 0) then
        return (positiveContributions − negativeContributions)/(totalContirbutions)
    end if
    return 0
end function
```

Algorithm 1 outlines the calculation process for the reputation score. The stance of a participant regarding a news can either be 1 for "Support" or -1 for "Deny", while news status can either be 1 for "True", − 1 for "False" or 0 for "Disputed". Consequently, the resulting reputation score ranges from − 1 to 1, with the magnitude indicating the participant's perceived trustworthiness.

Given the impact of stances on both reputation and credibility scores, we advocate presenting the stance classification results to participants. This approach not only enhances transparency and engagement, but also provides a foundation for reinforcement learning and empowers participants to adjust classification errors.

3.2.2 Credibility score

The credibility score assesses the reliability of a news based on participants' interactions and consensus. Similar to reputation scores, credibility scores are not explicitly stored but calculated on the fly, by only considering the stances of reputable participants (*i.e.* participants with a positive reputation) and their respective reputation scores. This participatory reputation-based approach enhances the system's ability to identify and combat fake news by leveraging the wisdom of the crowd and giving greater weight to trusted participants. The credibility score of a news, denoted C, is calculated using the following formula:

$$C = \frac{\sum_{i=1}^{N}(\text{Stance}_i \times \text{Reputation}_i)}{N}, \tag{1}$$

where N is the total number of participants with positive reputations engaging with the news, $\text{Stance}_i \in \{\text{-1,1}\}$ and $\text{Reputation}_i \in [-1, 1]$ are respectively the stance and the reputation score of participant i.

The above formula yields a value $C \in [-1, 1]$. This normalized score serves as an indicator of the overall trustworthiness of the news article. A positive score suggests a general consensus in favor of the news, while a negative score indicates a general consensus against the news. The magnitude of the score reflects the strength of this consensus.

3.3 Hoeffding bound—based news credibility assessment

The Hoeffding Bound [20] states that, with probability $1 - \delta$, the true average of a random variable of range R will not differ from the estimated average after N independent observations by more than :

$$\epsilon = \sqrt{\frac{R^2 \ln(1/\delta)}{2N}} \tag{2}$$

The Hoeffding bound is useful because it is assumption-free and holds true regardless of the distribution generating the values, and depends only on the range of values R, the number of observations N and the desired confidence $1 - \delta$. With R and δ fixed, the only variable left to change the Hoeffding bound ϵ is the number of observations N. As N increases, ϵ will decrease, in accordance with the estimated average getting ever closer to its true value. From a practical point of view, the confidence level $\delta \in [0, 1]$ allows to fine-tune the "acceptable" margin of error ϵ.

For the purpose of deciding on the status of a news, the random variable being estimated is $p = \text{Stance} \times \text{Reputation}$, the stance of a participant regarding a news, weighted by its reputation score. The average of p corresponds to $C = \frac{\sum_{i=1}^{N}(\text{Stance}_i \times \text{Reputation}_i)}{N}$, the credibility score of the news. Since the *Stance* can either be 1 or -1 and *Reputation* ranges from -1 to 1, C ranges from -1 to 1, meaning that $R = 2$.

When $C - \epsilon > 0$, this implies that C is confidently positive, and the news can be considered "True": *i.e.*, we can assert with a confidence level of $1 - \delta$ that C will

always remain positive, and hence that the news will remain "True" when additional stances become available. Similarly, when $C + \epsilon < 0$, this implies that C is confidently negative, and we can assert, with a confidence level of $1 - \delta$, that the news is "False". As shown in the flowchart of Fig. 1, when the conditions making a news either "True" or "False" are not met, the news is considered to be in a "Disputed" state.

Example 1 Let's consider the following stances regarding a claim and suppose that $\delta = 0.25$.

i	Stance$_i$	Reputation$_i$
1	1 (Support)	0.1
2	− 1 (Deny)	0.9
3	− 1 (Deny)	0.9

According to Eqs. 1 and 2, the credibility score and the margin of error are resp. −0.57 and 0.96. Since $|C| < \epsilon$, we cannot conclude that the news is False and it remains in a "Disputed" state. The relatively large value of the margin of error ϵ is essentially due to the low number of participants.

Example 2 Let's suppose that 2 more participants commented the news as follows.

i	Stance$_i$	Reputation$_i$
1	1 (Support)	0.1
2	− 1 (Deny)	0.9
3	− 1 (Deny)	0.9
4	− 1 (Deny)	0.9
5	− 1 (Deny)	0.9

In this example $C = -0.7$, and $\epsilon = 0.69$. Since $C + \epsilon < 0$, we conclude, with a confidence level of 75% (corresponding to $1 - \delta$), that C will always remain negative as new stances become available and consider the news as "False."

4 Experimental study

4.1 Tools and dataset

One of the major difficulties we faced in our experimental study was the lack of datasets covering all aspects of stance-based fake news detection. Existing fake news detection datasets do not yet support a fully automated end-to-end setup, and often treat stance detection, source credibility, and news classification as separate tasks. This explains why the majority of existing blockchain-based fake news detection approaches primarily concentrate on attack scenarios rather than the system's efficiency in identifying fake news.

Despite the fact that the RumourEval dataset [21] does not associate comments with the corresponding participants, and, hence, does not allow us to track participants' good and bad contributions, we used it in our experiments because it is one of the very few datasets that provides social media conversations annotated for both stance and veracity. The dataset contains 3342 conversations from Twitter covering a variety of topics, including politics, natural disasters, and celebrity gossip. RumourEval contains two subtasks, "Subtask A—Stance Detection" and "Subtask B—Veracity Detection". The goal of Subtask A is to classify the stance of a social media post towards a rumored claim (SDQC: Support, Deny, Query, Comment). The goal of subtask B is to classify the veracity of a rumored claim (True, False, Undetermined). Figure 2 provides an example retrieved from the dataset. Due to class imbalance, the RumourEval 2019 benchmark [21] adopts the macro-averaged F1 metric to evaluate competing approaches. The relatively low F1-scores achieved by most competing approaches (Table 4) underscore the non-trivial nature of the task. The benchmark creators [21] offer baseline models for both subtasks. For Subtask A, they provide a Keras implementation of BranchLSTM [22], the winning system of RumourEval 2017 Subtask A. Utilizing the conversation's structure, BranchLSTM divides it into linear branches and employs LSTM layers to process tweet sequences, generating a stance label at each time step. For subtask B, [21] modified BranchLSTM to produce a single output for each branch. The veracity prediction for the thread is then determined by majority voting over per-branch outcomes [21]. In this study, we configured our framework using BranchLSTM [22] as the stance detector, implemented the core logic using a Solidity 0.9.0 [23] smart contract, and conducted testing using Ganache [24].

For the purpose of tracking participants' actions, we simulated 42 participants,[1] generating approximately the same number of stances. Randomly assigning comments to participants could lead to borderline low magnitudes reputations (*i.e.*, close to zero). Such random assignment fails to accurately mirror real-world scenarios where participants exhibit varying degrees of reliability and where we encounter a mix of reputable participants, malicious actors, and those falling in between. Accordingly, to introduce diversity and capture the heterogeneity present in real-world social networks, we categorized participants into

[1] With 41 being the highest number of comments on one tweet in the dataset.

Fig. 2 Example extracted from the RumourEval dataset [21]

u1: These are not timid colours; soldiers back guarding Tomb of Unknown Soldier after today's shooting #StandforCanada PICTURE **[support]**
u2: @u1 Apparently a hoax. Best to take Tweet down. [deny]
u3: @u1 This photo was taken this morning, before the shooting. [deny]
u4: @u1 I dont believe there are soldiers guarding this area right now. [deny]
u5: @u4 wondered as well. Ive reached out to someone who would know just to confirm that. Hopefully get response soon. [comment]
u4: @u5 ok, thanks. [comment]

three reputation score ranges: lower than -0.5, higher than 0.5, and between -0.5 and 0.5. These categories represent, respectively, malicious participants, reputable participants, and borderline participants. We also explored varying fractions within each category.

4.2 Results and discussion

4.2.1 Results

The primary objective of our experimental study is to assess the overall effectiveness of our reputation-driven, stance-based fake news detection framework and to identify the conditions necessary for its success. To this end, we conducted our experiments with four specific objectives: (*i*) illustrating and quantifying the impact of the efficiency of the stance detector on the system performance; (*ii*) examining the influence of varying fractions of malicious and reputable participants; (*iii*) exploring the system sensitivity to variations in δ, the confidence level associated with the Hoeffding bound; and (*iv*) analyzing the impact of the length of participants' historical trajectories.

To illustrate the impact of the stance detector on overall performance, we compare the results obtained using the BranchLSTM stance detector [22] (Table 1) to those that would be achieved when all stances are correctly determined (Table 2). The latter scenario corresponds to a voting system where participants explicitly provide their stances on news. It's worth mentioning that such a scenario is covered by our framework, as participants have the ability to correct the detected stances. Tables 1 and 2 show that improving the performance of the stance detector module has the potential to yield an average improvement of approximately 12.2‘% in fake news detection.

To evaluate the influence of different participant distributions on the framework's performance, we examined four scenarios represented by distributions $Dist_1$ to $Dist_4$: $Dist_1$: [0.05 *Reputable*, 0.9 *Borderline*, 0.05 *Malicious*], $Dist_2$ [0.1 *Reputable*, 0.8 *Borderline*, 0.1 *Malicious*], $Dist_3$: [0.15 *Reputable*, 0.7 *Borderline*, 0.15 *Malicious*], and $Dist_4$: [0.2 *Reputable*, 0.6 *Borderline*, 0.2 *Malicious*]. As expected and reflected in Tables 1 and 2, the framework effectiveness is not impacted by the fraction of malicious participants. However the effectiveness is highly influenced by the fraction of reputable participants, with poor performance observed at very low fractions (*e.g.* $Dist_1$). As shown in Tables 1 and 2, a 5% increase in the fraction of reputable participants results in an average 8.75% improvement in the F1-score obtained by the system.

As mentioned earlier, the Hoeffding bound is known for its conservative nature. This characteristic is confirmed in our experimental study, where relatively low values of the confidence level $(1-\delta)$ result in good performance. This is attributed to the fact that when δ is low the Hoeffding bound refrains from making decisions regarding the status of news until it reaches a high level of confidence. Consequently, a significant portion of news remains categorized as "Disputed" rather than being identified as "True" or "False". As demonstrated in Tables 1 and 2, regarding this specific dataset, the system's performance stabilizes or exhibits marginal improvement starting from a $\delta = 0.4$.

The continuous cycle of interaction, stance classification, reputation and credibility scores updates strengthens the overall effectiveness of our framework over time. However, a common challenge for crowdsourcing and user-generated data systems is the cold start problem, arising when there are few participants and stances during the initial stages. To evaluate the impact of the length of participants' interaction history on system performance, we considered scenarios where only 50% and 75% subsets (randomly selected) of the training dataset are used to derive the initial reputation scores. As shown in Table 3, the system's performance improves significantly as it matures, with participants experiencing less fluctuations in their reputation scores.

4.2.2 Discussion

Our framework demonstrates promising results on the RumourEval 2019 benchmark. However, as is often the case, benchmarks may not fully capture the complexities of the real world. This underscores the importance of considering the factors discussed below.

The findings from our experimental study have shed light on the key factors influencing the efficiency of our framework. In particular, results revealed that, while participants with low reputations have negligible influence on the system's performance, the overall effectiveness hinges critically on the proportion of reputable participants. This reliance can be attributed to two factors: the Proof of

Table 1 Macro-averaged F1 achieved by our framework using BranchLSTM [22] as Stance Detector

δ	$Dist_1 : [0.05, 0.9, 0.05]$ (%)	$Dist_2 : [0.1, 0.8, 0.1]$ (%)	$Dist_3 : [0.15, 0.7, 0.15]$ (%)	$Dist_4 : [0.2, 0.6, 0.2]$ (%)
0.2	15.63	27.75	30.59	34.07
0.3	19.74	31.47	37.88	38.55
0.4	24.37	34.59	40.77	49.83
0.5	28.72	37.57	42.58	50.22
0.6	28.72	37.90	49.29	50.22

Table 2 Macro-averaged F1 achieved by our framework when all stances are correctly identified (*i.e.* similar to voting)

δ	$Dist_1 : [0.05, 0.9, 0.05]$ (%)	$Dist_2 : [0.1, 0.8, 0.1]$ (%)	$Dist_3 : [0.15, 0.7, 0.15]$ (%)	$Dist_4 : [0.2, 0.6, 0.2]$ (%)
0.2	21.64	30.40	33.28	41.49
0.3	30.50	37.15	39.90	51.97
0.4	39.20	41.03	49.29	78.20
0.5	40	49.29	54.83	78.20
0.6	40	49.29	67.04	78.20

Table 3 Macro-averaged F1 score as a function of the fraction of the training dataset used to derive initial reputations ($\delta = 0.5$)

Fraction	$Dist_1 : [0.05, 0.9, 0.05]$ (%)	$Dist_2 : [0.1, 0.8, 0.1]$ (%)	$Dist_3 : [0.15, 0.7, 0.15]$ (%)	$Dist_4 : [0.2, 0.6, 0.2]$ (%)
Proposed approach with BranchLSTM stance detector				
0.5	9.97	13.65	18.29	22.36
0.75	17.52	22.31	26.24	31.74
Proposed approach with voting				
0.5	18.70	24.96	31.03	38.48
0.75	27.14	36.22	43.72	54.62

Reputation mechanism, which filters out stances from low-reputation participants, and the Hoeffding bound, which requires a minimum level of corroborating evidence (support) before reaching a verdict on news veracity. This highlights a potential weakness in scenarios with sparse reliable contributors or during the initial stages, as the system may struggle in reaching a consensus, leaving a significant portion of news in "Disputed" status.

Results also showed to what extent the efficiency of the stance detector, represented by the BranchLSTM model, influences the system's overall performance (with an observed average potential improvement of approximately 12.2% in fake news detection). This underscores the need for further exploration of more sophisticated stance detectors, especially as stance detection in real-world scenarios may face challenges, such as sarcasm and evolving language patterns.

Regarding the Hoeffding bound, its conservative nature introduces a trade-off between confidence levels and precision. As show in our experimental study, lowering the confidence level prompts the Hoeffding bound to make decisions, resulting in more decisive judgments and improved F1-scores (as fewer news items are classified as "Disputed"). However, this also increases the risk of misclassifications, especially in situations with limited data or high uncertainty. While this adaptability allows the framework to be fine-tuned based on specific application requirements, it also prompts considerations regarding the balance between decisiveness and precision.

As shown in Table 3, the effectiveness of our framework is impacted by the length of participants' historical stance data. To mitigate this cold start problem, two "airdrop" strategies (using reputation as a token) deserve further exploration: (*i*) Uniform seeding: similar to some existing cryptocurrency blockchains, a (low) fixed reputation score is uniformly distributed to new participants; and (*ii*) Weighted seeding: high initial reputation scores are assigned to trusted entities such as fact-checking

Table 4 Macro-averaged F1 achieved by the baseline model and the best-performing models

Approach	Score (%)
Baseline Majority Class [21]	22.41
BranchLSTM [22]	49.29
eventAI [25]	57.65
Fine-tuned Longformer [26]	58.68
Shared Multi-channel Interactions (MTL-SMI) [27]	68.5
Proposed approach with BranchLSTM stance detector ($\delta = 0.5$)	up to 50.22
Proposed approach with voting ($\delta = 0.5$)	up to 78.20

organizations, governmental press agencies, and academic institutions. These "seeded" participants would serve as anchors for the reputation system during the initial phase, providing a foundation for credibility assessment until new participants establish their reputation through interaction.

While achieving comparable performance to machine learning approaches (Table 4), our framework stands out due to its fully decentralized and scalable nature, coupled with its dynamic reputation-based news credibility assessment. This dynamic approach offers a more nuanced evaluation of news veracity compared to static ML models and fosters an environment where participant influence aligns with their established reliability. These features are achieved without relying on a central authority or requiring frequent model retraining to update a frozen knowledge of the world. This combination positions our framework as a promising solution in the complex landscape of fake news detection.

Finally, it is worth to noticing that the proposed framework and conventional ML-based approaches are not mutually exclusive. The reputation and credibility scores derived within our framework can serve as valuable inputs to ML models. These scores provide additional contextual information that has the potential to enhance the performance of ML algorithms in detecting fake news.

5 Conclusion

This paper introduces a novel framework that integrates stance detection and blockchain technology for a decentralized, scalable and collaborative fact-checking. The proposed reputation-driven system operates autonomously, eliminating the need for human intervention or reliance on a central authority. Within this framework, stance detection acts as an automated means of opinion retrieval regarding a claim, facilitating evidence gathering. The Proof of Reputation mechanism fosters an environment where reputable participants have a greater influence in shaping credibility judgments, ensuring that the evaluation of news benefits from the collective wisdom of high-reputation contributors and minimizing the potential impact of malicious actors. The incorporation of the Hoeffding bound provides statistical guarantees for news label assignment, addressing challenges posed by evolving data and ensuring robustness in dynamic environments. While allowing comparable results with approaches based on conventional (batch) machine learning models, the proposed framework stands out by its authority-free nature and by limiting the need for periodic retraining to update a model's static knowledge of the world.

Our envisioned future directions encompass several key aspects. Firstly, we aim to dynamically adjust the confidence level δ in the Hoeffding Bound, to ensure that the system's certainty in credibility assessments adapts to evolving contexts and dynamic news interactions. Additionally, we plan to introduce fine-grained categories beyond the binary true/false classification, such as "half-true" and "doubtful", etc. for a more nuanced analysis, particularly valuable for borderline cases. Furthermore, we envision leveraging external knowledge sources, including fact-checking databases, to enrich the system's evidence base and context. Finally, we intend to investigate adversarial attacks and test the system's robustness against potential manipulation attempts, such as coordinated bot campaigns or fake accounts.

Author contributions I.S and K.J drafted the main manuscript, I.S.,K.J., and C.A.P carried out the experimental study. O.K., K.J., and C.A.P. supervised and ensured the validation of each milestone. O.K. reviewed and proofread the manuscript.

Funding The authors have not disclosed any funding.

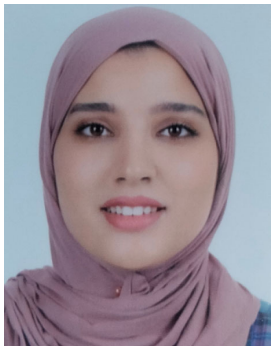
Data availability No datasets were generated or analysed during the current study.

Declarations

Competing interests The authors declare no competing interests.

References

1. Salah, I., Jouini, K., Korbaa, O.: Augmentation-based ensemble learning for stance and fake news detection. In: Advances in Computational Collective Intelligence - 14th International Conference, ICCCI 2022 Proceedings. Communications in Computer and Information Science, vol. 1653, pp. 29–41. Springer, Hammamet, Tunisia (2022). https://doi.org/10.1007/978-3-031-16210-7_3
2. Salah, I., Jouini, K., Korbaa, O.: On the use of text augmentation for stance and fake news detection. J. Inf. Telecommun. **7**(3), 359–375 (2023). https://doi.org/10.1080/24751839.2023.2198820
3. Slovikovskaya, V.: Transfer learning from transformers to fake news challenge stance detection (fnc-1) task. In: Proceedings of the 12th Language Resources and Evaluation Conference, pp. 1211–1218. European Language Resources Association, Marseille, France (2019). https://www.aclweb.org/anthology/2020.lrec-1.152
4. Alhassani, Z., Obaid, A.: A systemic literature overview of fake news challenge (fnc-1) dataset and its use in fake news detection schemes. J. Dis. Math. Sci. Cryptogr. **26**, 1197–1206 (2023). https://doi.org/10.47974/JDMSC-1567
5. Jemili, F., Meddeb, R., Korbaa, O.: Intrusion detection based on ensemble learning for big data classification. Cluster Comput. (2023). https://doi.org/10.1007/s10586-023-04168-7
6. Khan, A.U., Javaid, N., Khan, M.A., Ullah, I.: A blockchain scheme for authentication, data sharing and nonrepudiation to secure internet of wireless sensor things. Cluster Comput. **26**(2), 945–960 (2023). https://doi.org/10.1007/s10586-022-03722-z
7. Zarrin, J., Phang, H.W., Saheer, L.B., Zarrin, B.: Blockchain for decentralization of the internet: prospects, trends, and challenges. Cluster Comput. **24**(4), 2841–2866 (2021). https://doi.org/10.1007/s10586-021-03301-8
8. ANSAcheck. https://www.ansa.it/sito/static/ansa_check.html. Accessed 31 Oct 2023
9. New York Times News Provenance Project. https://newsprovenanceproject.com. Accessed 31 Oct 2023
10. Avelino, M., Rocha, A.A.D.A.: Blockproof: a framework for verifying authenticity and integrity of web content. Sensors (2022). https://doi.org/10.3390/s22031165
11. Wang, X., Xie, H., Ji, S., Liu, L., Huang, D.: Blockchain-based fake news traceability and verification mechanism. Heliyon **9**(7), 17084 (2023). https://doi.org/10.1016/j.heliyon.2023.e17084
12. Chen, C., Du, Y., Peter, R., Golab, W.M.: An implementation of fake news prevention by blockchain and entropy-based incentive mechanism. Soc. Netw. Anal. Min. **12**(1), 114 (2022). https://doi.org/10.1007/S13278-022-00941-5
13. Soprano, M., Roitero, K., La Barbera, D., Ceolin, D., Spina, D., Mizzaro, S., Demartini, G.: The many dimensions of truthfulness: crowdsourcing misinformation assessments on a multidimensional scale. Inf. Process. Manage. (2021). https://doi.org/10.1016/j.ipm.2021.102710
14. Yilmaz, T., Ulusoy, O.: Modeling and mitigating online misinformation: a suggested Blockchain approach (2023). https://doi.org/10.48550/arXiv.2303.10765
15. Kittur, A., Chi, E.H., Suh, B.: Crowdsourcing user studies with mechanical turk. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI '08, pp. 453–456. Association for Computing Machinery, New York, NY, USA (2008). https://doi.org/10.1145/1357054.1357127
16. Bhuiyan, M.M., Zhang, A.X., Sehat, C.M., Mitra, T.: Investigating differences in crowdsourced news credibility assessment: raters, tasks, and expert criteria. Proc. ACM Hum.-Comput. Interact. (2020) https://doi.org/10.1145/3415164
17. Denaux, R., Merenda, F., Gómez-Pérez, J.M.: Towards crowdsourcing tasks for accurate misinformation detection. In: Joint Proceedings of Workshops AI4LEGAL2020, NLIWOD, PROFILES 2020, QuWeDa 2020 and SEMIFORM2020 Colocated with the 19th International Semantic Web Conference (ISWC 2020), Virtual Conference, November, 2020. CEUR Workshop Proceedings, vol. 2722, pp. 159–167. CEUR-WS.org. https://ceur-ws.org/Vol-2722/semiform2020-paper-2.pdf
18. Marx, J., Cheong, M.: Decentralised social media: scoping review and future research directions. In: Australasian Conference on Information Systems (2023)
19. Boot, A.B., Dijkstra, K., Zwaan, R.A.: The processing and evaluation of news content on social media is influenced by peer-user commentary. Human. Soc. Sci. Commun. **8**(1), 209 (2021). https://doi.org/10.1057/s41599-021-00889-5
20. Hoeffding, W.: Probability inequalities for sums of bounded random variables. J. Am. Stat. Assoc. **58**, 13–30 (1963). https://doi.org/10.1080/01621459.1963.10500830
21. Gorrell, G., Bontcheva, K., Derczynski, L., Kochkina, E., Liakata, M., Zubiaga, A.: RumourEval 2019: Determining rumour veracity and support for rumours (2018). https://doi.org/10.48550/arXiv.1809.06683
22. Kochkina, E., Liakata, M., Zubiaga, A.: All-in-one: Multi-task learning for rumour verification. In: Proceedings of the 27th International Conference on Computational Linguistics, pp. 3402–3413. Association for Computational Linguistics, Santa Fe, New Mexico, USA (2018). https://doi.org/10.48550/arXiv.1806.03713
23. Solidity. https://soliditylang.org/. Accessed 31 Dec 2023
24. Ganache. https://www.trufflesuite.com/ganache. Accessed 31 Dec 2023
25. Li, Q., Zhang, Q., Si, L.: eventAI at SemEval-2019 task 7: Rumor detection on social media by exploiting content, user credibility and propagation information. In: May, J., Shutova, E., Herbelot, A., Zhu, X., Apidianaki, M., Mohammad, S.M. (eds.) Proceedings of the 13th International Workshop on Semantic Evaluation, pp. 855–859. Association for Computational Linguistics, Minneapolis, Minnesota, USA (2019). https://doi.org/10.18653/v1/S19-2148
26. Khandelwal, A.: Fine-tune longformer for jointly predicting rumor stance and veracity. CODS-COMAD '21, pp. 10–19. Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3430984.3431007
27. Liu, Y., Yang, X., Zhang, X., Tang, Z., Chen, Z., Zheng, L.: Predicting rumor veracity on social media with cross-channel interaction of multi-task. Neural Comput. Appl. (2024). https://doi.org/10.1007/s00521-024-09519-y



Ilhem Salah is currently pursuing a Ph.D. in Computer Science at Sousse University (Tunisia) after earning her Master's degree in Distributed Computing from the same institution. Her research primarily focuses on fake news detection in social media, as well as machine learning, deep learning, distributed ledgers, and natural language processing.

Khaled Jouini received the Ph.D. degree in Computer Science from Paris-Dauphine University (France) in 2008. He was a research staff member at Telecom ParisTech (France). Since 2011, he has been with University of Sousse (Tunisia), where he is currently an Associate Professor. His research interests include data engineering, natural language processing, and large-scale data management and mining.

Cyril-Alexandre Pachon received his Ph.D. in Computer Science, Systems, and Communication from Université Joseph Fourier (France) in 2005. He is currently the Director of Studies at École Hexagone (France). Previously, he spent over a decade at SUPINFO International University (France), where he managed the Robotics Laboratory and led initiatives in robotics competitions. His research interests include robustness testing, test case generation, data science, and artificial intelligence applications.

Ouajdi Korbaa is a full-time professor at the University of Sousse (Tunisia). He received his Engineering Diploma from the Ecole Centrale de Lille (France) in 1995 and his Master's degree in Production Engineering and Computer Science from the University of Lille (France) in the same year. He obtained his Ph.D. in Production Management, Automatic Control, and Computer Science from the University of Science and Technologies of Lille (France) in 1998. Pr. Korbaa has published approximately 190 research papers on optimization, applied and computational mathematics, manufacturing engineering, and computer engineering.

.

Q1.2 Detecting Unknown Intrusions from Large Heterogeneous Data through Ensemble Learning

Farah Jemili, Khaled Jouini & Ouajdi Korbaa
Intelligent Systems with Applications. Vol. 25, 200465, March 2025.
ISSN : 26673053, Elsevier B.V.

DOI : https://doi.org/10.1016/j.iswa.2024.200465
SJR best quartile : Q1, SJR : 0.96.

Intelligent Systems with Applications

COUNTRY	SUBJECT AREA AND CATEGORY	PUBLISHER	H-INDEX
Netherlands Universities and research institutions in Netherlands Media Ranking in Netherlands	Computer Science Artificial Intelligence Computer Science Applications Computer Science (miscellaneous) Computer Vision and Pattern Recognition Signal Processing	Elsevier B.V.	18

PUBLICATION TYPE	ISSN	COVERAGE	INFORMATION
Journals	26673053	2021-2023	Homepage How to publish in this journal

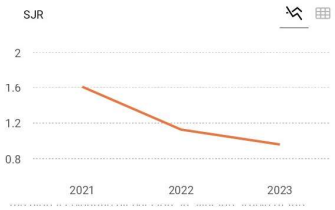
SCOPE

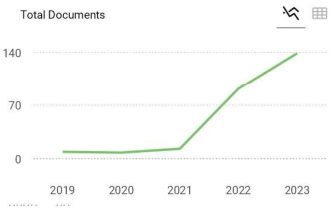
Intelligent Systems with Applications (ISWA) is a peer reviewed open-access journal which focuses on the achievements of research and applications related to intelligent systems (IS). IS can be applied to all aspects of human enterprize, such as business & finance, manufacturing & supply chains, agriculture, transportation, engineering, medicine and health, education, entertainment, culture, travel, media, the Internet, etc. ISWA covers a broad spectrum of applications in the community, including industry, government, and academia. The journal aims to publish papers dealing with, but not limited to, the following research fields: Knowledge Representation and Reasoning, Machine Learning (ML) and Neural Computing, Evolutionary Computation, Fuzzy Systems, Intelligent Information Processing, Intelligent Control and Robotics, Multi-agent Systems and Programming. The journal welcomes submissions of IS-applications in various areas: Intelligent Cities, Industries, Consuming, Medical Treatment and Health, Agriculture, Business and Finance, Internet of Things (IoT). Research addressing IS-applications in other fields is also encouraged. Submissions must be novel, technically sound, and clearly presented. ISWA accepts both regular papers and survey articles. Submissions meeting journal criteria will undergo a single-blind review process, utilizing a minimum of two (2) external referees. Our dedicated editorial team, together with active researchers from related fields, will ensure that papers move through the evaluation and review as rapidly as possible without compromising on the quality of the process.

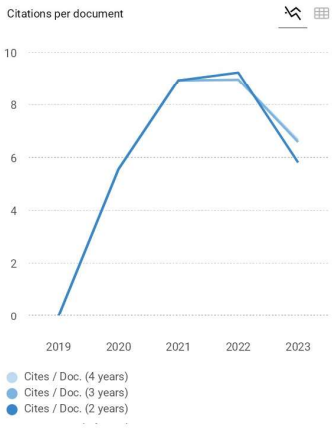
FIND SIMILAR JOURNALS

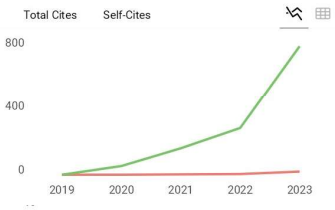
options

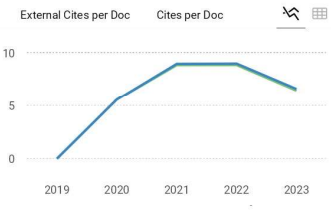
1	2	3	4	5
Applied Artificial Intelligence	**PeerJ Computer Science**	**Computational Intelligence**	**Expert Systems**	**Journal of Advances in Information Technology**
GBR	USA	GBR	GBR	USA
84% similarity	**83%** similarity	**81%** similarity	**81%** similarity	**80%** similarity

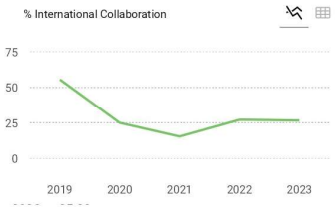


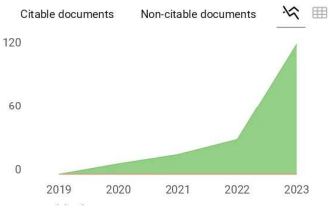


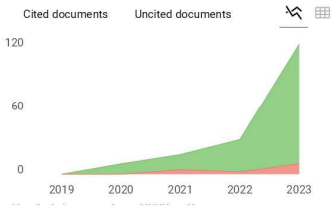


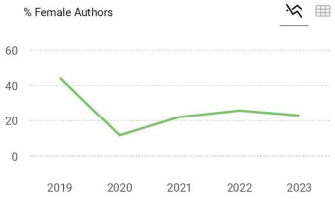


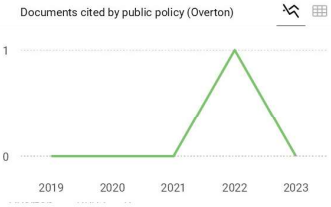


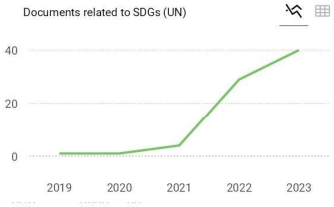












← Show this widget in your own website

Just copy the code below and paste within your html code:

<a href="https://www.scim

SCImago Graphica

Explore, visually communicate and make sense of data with our **new data visualization tool**.

Metrics based on Scopus® data as of March 2024

----- Message transféré -----
De : "Elsevier - Author Forms" <oasupport@elsevier.com>
À : "jmili_farah@yahoo.fr" <jmili_farah@yahoo.fr>
Cc :
Envoyé : mar., déc. 10, 2024 à 13:40
Objet : Rights and Access form completed for your article [ISWA_200465]
Elsevier

cover
Dear Dr Jemili,

Thank you for completing the Rights and Access Form for your article Detecting Unknown Intrusions from Large Heterogeneous Data through Ensemble Learning on December 10, 2024.

The Order Summary is attached to this email. If you wish to make any changes, please contact our Researcher Support team immediately through one of the contact options mentioned on the support site: https://service.elsevier.com.

Your article is free for everyone to read online at https://doi.org/10.1016/j.iswa.2024.200465

If you have any questions, please do not hesitate to contact us. Quote our article reference ISWA_200465 in all correspondence.

Now that your article has been accepted, you will want to maximize the impact of your work. Elsevier facilitates and encourages authors to share their article responsibly. To learn about the many ways in which you can share your article while respecting copyright, visit: www.elsevier.com/sharing-articles.

Kind regards,
Elsevier Researcher Support

Image Seven strategies for you to create a brand and promote your research
Learn how to give your research the visibility it deserves with these seven strategies.

> Access module now

Have questions or need assistance?
Please do not reply to this automated message.
For further assistance, please visit our Elsevier Support Center where you can search for solutions on a range of topics and find answers to frequently asked questions. From here you can also contact our Researcher Support team via 24/7 live chat, email or phone support.

ELSEVIER

Rights and Access

Detecting Unknown Intrusions from Large Heterogeneous Data through Ensemble Learning

Corresponding author	Dr Farah Jemili
E-mail address	jmili_farah@yahoo.fr

Journal	Intelligent Systems with Applications
Article number	200465
Our reference	ISWA_200465
PII	S2667-3053(24)00139-X
DOI	10.1016/j.iswa.2024.200465

Order Confirmation

Thank you for taking the time to complete the Rights and Access form.

Order number
OACSRISWA2004650

Order date
10 December 2024

Publishing Option
Open access

User License
CC BY-NC 4.0

Publishing Agreement

- I am one author signing on behalf of all co-authors of the manuscript and I am duly authorized to do so by all co-authors of the manuscript
- I am signing on behalf of the corresponding author.
  - **Name/Job title/Company:** Farah Jemili
  - **E-mail address:** jmili_farah@yahoo.fr

Important information

Availability and sharing of your article
Your article has been made immediately available to everyone. It can be shared and reused in the ways described by the Creative Commons end user license you have selected.

Institutional and funder agreements
Any fees covered under agreements are subject to institution or funder approval.

Payment of open access fees
Payment is your responsibility. If we invoice your institution and don't receive payment, or if fees covered under agreements are not approved, we will reissue the invoice to you.

You have agreed to the above and to the full Elsevier terms and conditions of purchase for open access services.

Total payment due

Price (excluding taxes)	
	USD 1,500.00
Discount	
	- 1,500.00
To pay	
	0.00
Total price ex. Tax	

Intelligent Systems with Applications 25 (2025) 200465

Contents lists available at ScienceDirect

Intelligent Systems with Applications

journal homepage: www.journals.elsevier.com/intelligent-systems-with-applications

Detecting unknown intrusions from large heterogeneous data through ensemble learning

Farah Jemili *, Khaled Jouini, Ouajdi Korbaa

Universite de Sousse, ISITCom, MARS Research Laboratory, LR17ES05, Hammam Sousse, 4011, Tunisia

ARTICLE INFO

Keywords:
Big heterogeneous data
Intrusion detection
Data fusion

ABSTRACT

The rapid expansion of data volumes, technological advancements, and the emergence of the Internet of Things (IoT) have heightened concerns regarding the detection of unknown intrusions based on singular sources of network traffic. This progression has led to the generation of vast and diverse datasets originating from various sources including IoT devices, web applications, and web services. Effectively discerning attacks within such a heterogeneous network traffic landscape necessitates the identification of underlying security behaviors, essential for developing an efficient analysis information system.

This paper aims to establish a comprehensive framework for network intrusion detection. The proposed methodology involves the synthesis of network features into a universal security database through the utilization of Term Frequency-Inverse Document Frequency Terms (TF-IDF) and semantic Cosine similarity. By amalgamating a diverse array of data flows, a set of universal features is generated, facilitating storage within the newly devised universal representation. Subsequently, Principal Component Analysis (PCA) is employed to reduce the dimensionality of the extensive universal security database while preserving essential information. Leveraging Ensemble Learning, a novel method is introduced for the detection of unknown attacks.

The efficacy of the developed database is evaluated using various Machine Learning algorithms, including Naïve Bayes, K-Nearest Neighbor, Logistic Regression, Decision Tree, and Random Forest. Furthermore, Ensemble Learning methods are assessed under two distinct scenarios. Experimental findings, conducted on datasets such as CICIDS 2017, NSL-KDD, and UNSW, demonstrate the universality, versatility, and effectiveness of the proposed approach, particularly in accommodating datasets with diverse structures.

1. Introduction

Intrusion Detection Systems (IDS) have long been essential tools for identifying potential attacks within computer networks and systems. Recently, the integration of Machine Learning (ML) algorithms has further enhanced IDS by enabling more sophisticated data analysis and classification for intrusion detection. In the context of ML, IDS function as classifiers, categorizing network traffic data and predicting potential attacks based on this classification (Patel, Taghavi, Bakhtiyari & Júnior, 2013).

However, IDS face increasing challenges due to the exponential growth of data and the proliferation of Internet of Things (IoT) devices, which generate diverse and complex data. While recent advancements in big data have led to the development of distributed architectures for IDS (Intrata, Grif & Dostovalov, 2021; Othman et al., 2018; Peng, Leung & Huang, 2018), the dynamic and heterogeneous nature of cyberspace complicates the detection of unknown intrusions across varied network data sources. Given the prevalence of interconnected computer systems in real-world applications, the traditional approach of creating separate IDS for each network type is increasingly impractical.

1.1. Motivation and contributions

- Motivation:
  - The rise in volume and diversity of network traffic, particularly with IoT, has amplified the challenge of detecting unknown attacks in real-time.
  - Conventional IDS struggle to process heterogeneous data from multiple sources, necessitating a universal framework for intrusion detection.
  - An adaptable, efficient, and scalable IDS approach is critical for real-world environments where threats constantly evolve.

* Corresponding author.
E-mail address: jmili_farah@yahoo.fr (F. Jemili).

https://doi.org/10.1016/j.iswa.2024.200465
Received 26 April 2024; Received in revised form 14 November 2024; Accepted 5 December 2024
Available online 9 December 2024

- Contributions of This Study:
  - Universal Security Database: We propose a comprehensive database that integrates features from diverse network types, creating a versatile foundation for IDS.
  - Universal Feature Representation: By leveraging Term Frequency-Inverse Document Frequency (TF-IDF) and Cosine similarity, our framework effectively fuses heterogeneous data sources.
  - Dimensionality Reduction: Principal Component Analysis (PCA) is applied to streamline the feature space, ensuring both efficiency and preservation of essential information.
  - Ensemble Learning Method: A unique ensemble method, combining parallel and sequential approaches, is introduced to enhance the detection accuracy of unknown attacks.
  - Comprehensive Evaluation: The proposed framework is rigorously tested on multiple scenarios using CICIDS 2017, NSL-KDD, and UNSW-NB15 datasets, demonstrating its effectiveness and adaptability.

1.2. Paper structure

The remainder of this paper is organized as follows: Section 2 reviews the current state of IDS and the ongoing challenges. Section 3 discusses network traffic sources and introduces a novel framework for data representation. Section 4 details the proposed methodology. Section 5 presents evaluation results, and Section 6 provides conclusions from the study.

2. Related works

Traditional machine learning (ML) tools, while effective in certain contexts, have struggled to keep pace with the ever-expanding diversity and sheer volume of network traffic data. As a result, researchers have increasingly turned to big data technologies as a means of enhancing intrusion detection capabilities. To tackle the challenge posed by the sheer magnitude of data, innovative approaches leveraging distributed architectures, distributed storage systems, and parallelization paradigms have been developed.

Peng, Leung and Huang (2018) introduced a method combining Mini Batch K-means with Principal Component Analysis (PCA) to differentiate between normal and malicious behavior in network traffic. This approach harnesses the power of distributed computing for scalability and efficiency, but it focuses on a single data source and does not explore ensemble learning methods for integrating multiple datasets. Our proposed approach advances this work by integrating data from NSL-KDD, UNSW, and CICIDS datasets and employing a novel ensemble learning strategy that combines both parallel and sequential methods for enhanced intrusion detection.

Elayni et al.'s survey (Elayni, Jemili, Korbaa & Soulaimen, 2019) provides valuable insights into the IDS process, spanning from data collection to detection methods, and highlights the importance of big data storage and analysis techniques. However, this survey does not address the specific challenges of data fusion across heterogeneous sources or the application of ensemble learning techniques. Our work builds upon these foundational concepts by proposing a comprehensive approach that integrates advanced feature extraction techniques with a unique ensemble learning framework designed to handle diverse data environments.

Distributed frameworks such as Hadoop and Spark have emerged as pivotal platforms for managing large volumes of network traffic data (Intrata, Grif & Dostovalov, 2021; Othman et al., 2018; Patel, Taghavi, Bakhtiyari & Júnior, 2013; Peng, Leung & Huang, 2018; Zuech, Khoshgoftaar & Wald, 2015). These frameworks are essential for efficient data processing but often focus on individual datasets rather than exploring techniques for fusing diverse data sources. Our approach extends these frameworks by applying a novel ensemble learning strategy that integrates multiple data sources and methods to improve detection performance.

Zhou et al. (2018) introduced a taxonomy that addresses the heterogeneity of network data, including host logs, application logs, and wireless network traffic. While this taxonomy provides a framework for understanding data diversity, it does not explore the integration of these diverse data types through advanced feature extraction and ensemble learning techniques. Our study contributes to this area by developing methods for effective data fusion and leveraging ensemble learning to address the complexities of heterogeneous network environments.

Recent studies such as ARGUS IDS and CIC Flow Meter (2022) have proposed advanced methods for feature extraction and classification, including multi-dimensional feature fusion and deep neural network-based reconstruction techniques. While these methods address feature extraction challenges, our work introduces a novel combination of data fusion techniques and ensemble learning approaches that significantly enhances the effectiveness of intrusion detection systems across a range of datasets and environments, (Table 1).

To further enhance intrusion detection in cybersecurity, the works in Arasteh et al. (2024), Arasteh, Bouyer, Sefati and Craciunescu (2024), and Danesh, Karimi and Arasteh (2024) offer noteworthy contributions focused on specialized threat detection. The paper (Arasteh, Bouyer, Sefati & Craciunescu, 2024) introduces a hybrid approach to SQL injection (SQLi) detection using the Binary Olympiad Optimizer (BOO) for feature selection combined with classification algorithms. This method efficiently reduces computational complexity and achieves high detection accuracy by narrowing down to relevant features for SQLi detection. Similarly, (Arasteh et al., 2024) addresses SQLi detection by employing the Binary Gray Wolf Optimizer (BGWO) alongside machine learning classifiers, constructing a dataset with essential SQLi indicators to improve model accuracy and reduce computational demands. While both approaches offer effective feature selection techniques that enhance traditional SQLi detection methods, they are limited to specific types of attacks within SQLi threats, and neither work expands to multiple types of network traffic data or addresses the scalability challenges associated with big data environments.

In Danesh, Karimi and Arasteh (2024), CMShark is introduced as a network-based, infrastructure-independent method for detecting crypto-jacking attacks. The study uses a hybrid approach, combining machine learning classification, IP blacklisting, and payload inspection, providing robust crypto-jacking detection through packet size classification, known IP addresses, and keyword-based payload inspection. Although CMShark's multi-faceted design improves crypto-jacking detection accuracy, its application is limited to network edge deployments and specific attack types, making it less versatile for large-scale, heterogeneous network environments.

While (Arasteh et al., 2024; Arasteh, Bouyer, Sefati & Craciunescu, 2024), and (Danesh, Karimi & Arasteh, 2024) introduce innovative and effective detection methods for SQLi and crypto-jacking threats, they primarily focus on isolated attack vectors, relying on specific feature selection optimizers and detection techniques that are not optimized for the integration of diverse data sources or scalable, distributed architectures. In contrast, our approach incorporates an ensemble learning framework that unites data from the NSL-KDD, UNSW, and CICIDS datasets, accommodating a broader spectrum of attacks. This ensemble learning strategy not only includes both parallel and sequential methods for enhanced intrusion detection but also addresses data fusion from diverse sources, surpassing the single-source limitations of the discussed studies. By leveraging distributed computing frameworks such as Hadoop and Spark, our approach achieves the scalability necessary to handle big data environments, a critical factor overlooked in Arasteh et al. (2024), Arasteh, Bouyer, Sefati and Craciunescu (2024), Danesh, Karimi and Arasteh (2024).

In summary, our contribution significantly advances the field by addressing the challenges of diverse data integration, scalability, and multi-model ensemble learning, delivering a more comprehensive and robust intrusion detection solution adaptable to complex and large-scale

Table 1
Related work.

Study	Key Contributions	Limitations	Our Novelty and Differentiation
Peng, Leung and Huang (2018)	Combines Mini Batch K-means with PCA for network traffic analysis	Focuses on a single data source; lacks ensemble methods	Integrates multiple datasets (NSL-KDD, UNSW, CICIDS) with ensemble learning (parallel and sequential) to improve detection accuracy
Elayni, Jemili, Korbaa and Soulaimen (2019)	Survey on IDS process from data collection to detection methods	Does not address data fusion or ensemble learning techniques	Proposes a comprehensive approach integrating data fusion and ensemble learning for diverse data environments
Distributed Frameworks (Patel, Taghavi, Bakhtiyari & Júnior, 2013; Intrata, Grif & Dostovalov, 2021; Peng, Leung & Huang, 2018; Othman et al., 2018; Zuech, Khoshgoftaar & Wald, 2015)	Utilizes Hadoop and Spark for efficient data processing	Often limited to individual datasets; lacks data fusion focus	Extends these frameworks with a novel ensemble learning strategy that integrates multiple data sources
Zhou et al. (2018)	Introduces taxonomy for network data heterogeneity	Does not explore data integration or advanced feature extraction techniques	Develops methods for effective data fusion and utilizes ensemble learning to handle heterogeneous network environments
ARGUS IDS	Proposes multi-dimensional feature fusion and stacking ensemble	Primarily focuses on feature extraction; limited data sources	Combines advanced feature extraction techniques with ensemble learning, enhancing effectiveness across various datasets
CIC Flow Meter	Novel reconstruction method using deep neural networks for feature extraction	Focuses on feature extraction and classification	Introduces a unique combination of data fusion and ensemble learning to significantly enhance intrusion detection capabilities

network environments.

Summary of Our Contributions:

- **Data Fusion:** Integrates multiple heterogeneous datasets (NSL-KDD, UNSW, CICIDS) to improve robustness and generalizability.
- **Ensemble Learning:** Utilizes a combination of parallel and sequential ensemble methods to enhance detection accuracy and performance.
- **Feature Extraction:** Employs innovative techniques like TF-IDF and Cosine similarity for effective feature extraction, addressing the unstructured nature of intrusion data.
- **Comprehensive Evaluation:** Conducts extensive experiments across diverse datasets to demonstrate the effectiveness and scalability of the proposed approach.

The proposed method addresses several key limitations commonly found in existing unknown intrusion detectors:

1. **Inability to Handle Heterogeneous Data**: Traditional detectors often focus on single-source or homogenous datasets, which limits their applicability in real-world scenarios where data sources are diverse. Our method overcomes this by creating a universal security database that integrates features from multiple, heterogeneous datasets (CICIDS 2017, NSL-KDD, UNSW-NB15).
2. **Limited Detection of Unknown Attacks**: Many intrusion detection systems rely on predefined patterns or signatures, making them less effective for unknown intrusions. By applying ensemble learning and a universal feature representation, our method enhances the detection capability for novel, previously unseen attacks.
3. **High Computational Requirements**: Existing systems sometimes struggle with large, high-dimensional data, which can slow down detection and reduce scalability. Our approach addresses this through Principal Component Analysis (PCA) for dimensionality reduction, which optimizes processing time and computational efficiency.
4. **Scalability and Adaptability Issues**: IDS often need extensive reconfiguration to adapt to different datasets or network environments. The proposed framework is designed to be adaptable across various data environments, increasing the scalability of the IDS.

3. Methodology formulation

3.1. Network data representation

This section delves into the intricacies of communication between network systems, as defined by the Open System Interconnection (OSI) reference model, which conceptualizes data transmission across seven layers from the physical layer to the application layer (ARGUS, IDS; CIC Flow, Meter). Data transmission occurs via packets, which serve as the fundamental unit of communication in network traffic (see Fig. 1). Each packet comprises a payload, representing the actual data being transmitted, along with one or more headers containing metadata such as source and destination IP addresses, service details, and protocols. Notably, headers play a crucial role in computer networking by providing essential information for accurate data transmission. For instance, packets conforming to the Telnet Network protocol (telnet) adhere to the TCP/IP standard and encompass a media access control (MAC) header, an internet protocol (IP) header, a transmission control protocol (TCP) header, a telnet header, and a telnet payload.

Intrusion Detection Systems (IDS) leverage headers from packets across various protocols to extract features specific to each type of network traffic data. For instance, tools like CICFlow Meter (CICIDS, 2017) are utilized to extract 76 distinct features from network traffic within datasets like CICIDS 2017.

However, two significant drawbacks are associated with the network traffic data utilized in IDS. Firstly, the detection focus is typically confined to individual packets or isolated data flows. While a single data flow may consist of multiple series of data packets from the same network type, real-world network traffic is characterized by heterogeneous data flows with diverse structural compositions. While heterogeneous data flow collection is recommended for IDS, it presents challenges for Machine Learning (ML) algorithms.

Secondly, metadata poses another challenge, as features are extracted from headers following universal network standards such as the OSI reference model and TCP/IP standard. However, proposed datasets often exhibit ambiguous feature names with differing notations, despite being derived from standard packet headers.

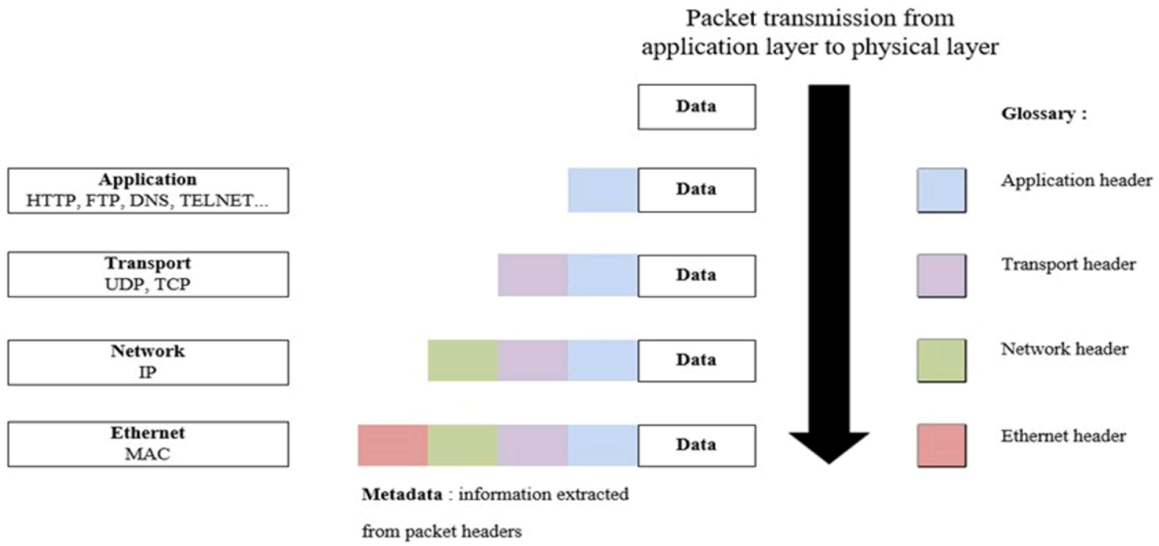


Fig. 1. The extraction of data and metadata from a packet.

In our study, we redefine the detection object as heterogeneous data flows, acknowledging the unstructured nature of the data. Our proposal centers on the development of a new universal representation encompassing all possible features established in accordance with network standards. This approach aims to address the complexities inherent in heterogeneous data environments and provide a comprehensive foundation for effective intrusion detection.

Our approach starts with understanding how data travels in a network, based on the Open System Interconnection (OSI) model. The OSI model describes data transmission in seven layers, from the physical layer to the application layer. Data is sent in packets, which are small units of data. Each packet has a payload (the actual data) and headers (metadata like source and destination IP addresses, and protocol information).

Intrusion Detection Systems (IDS) use these headers to extract features from network traffic. For example, tools like CICFlowMeter can extract 76 features from datasets such as CICIDS 2017. However, focusing only on individual packets or single data flows has two main problems:

1. Real-world network traffic is heterogeneous, meaning it contains data from various sources with different structures.
2. Metadata extracted from packet headers can have ambiguous names, even if they follow standard network protocols.

To address these issues, we propose a new method that handles heterogeneous data flows by creating a universal representation of network traffic.

3.2. Data flows reconstruction: big universal security database

In the realm of Machine Learning (ML), algorithms typically operate within the confines of specific data structures, rendering the utilization of a distinct Intrusion Detection System (IDS) for each network system impractical. This is primarily due to the interconnected nature of network layers, where an attack originating in the application layer can impact the physical layer, and vice versa. Therefore, an effective IDS necessitates the correlation and aggregation of data packets from various sources to provide comprehensive threat detection capabilities. In our proposed approach, we advocate for the development of a universal database capable of accommodating data from diverse sources.

The data, denoted as D, is sourced from various channels, as outlined in Section 3.1. Each data source is characterized by a specific structure, represented by a vector X. Data sharing the same vector structure is assigned to a common category, denoted as Ƙ. To establish a universal framework, we define a universal vector, U, encompassing all observable categories within network traffic, irrespective of its origin. The universal vector, U, comprises individual vectors, U= (X1, X2, …, Xk), with each vector defined by a set of features, F:

$$U_j = X_1 \leftarrow (F_1, F_2, \ldots, F_n)$$
$$X_2 \leftarrow (F_1, F_2, \ldots, F_m)$$
$$\vdots$$
$$X_k \leftarrow (F_1, F_2, \ldots, F_z)$$

Where $j = n + m + z$, representing the total dimensionality of the new universal vector, U.

At this juncture, our aim is to establish a standardized set of features within the universal vector, Uj. However, despite originating from the same network standard, certain features may exhibit redundancies or inconsistencies in naming conventions. In addressing this challenge, we leverage the metadata associated with each dataset, which provides a comprehensive description of all features. Our proposal entails the creation of a new corpus derived from metadata, wherein each feature is assigned a designation and detailed description. This corpus serves as a dynamic repository, automatically updated to accommodate new features as they are integrated. By harnessing metadata in this manner, we aim to streamline feature management and enhance the overall effectiveness of our universal IDS framework.

Machine Learning (ML) algorithms typically require specific data structures, making it impractical to use a separate IDS for each network system. This is because an attack in one layer (e.g., the application layer) can affect other layers (e.g., the physical layer). Therefore, our IDS needs to correlate data from various sources to detect threats effectively.

Our solution is to build a universal database that combines data from different sources. Here's how we do it:

1. **Data Collection**: We collect data (D) from various channels, each with a specific structure (vector X). Data with the same structure is grouped into a category (Ƙ).
2. **Universal Vector Creation**: We create a universal vector (U) that includes all categories of network traffic. The universal vector is a

combination of individual vectors ($U = X_1, X_2, \ldots, X_k$), each defined by a set of features (F).
3. **Standardization**: Although these features come from the same network standard, they might have different names or redundant information. We use metadata to create a new corpus that provides a clear description for each feature. This corpus is updated automatically as new features are added.

3.3. Construction of universal features vector

The universal features vector is a crucial component of our proposed methodology, designed to capture the essential characteristics of the input data for intrusion detection. Below, we provide a detailed description of the construction process, along with pseudocode to facilitate reproducibility.

1. Data Preprocessing:
   - Tokenize the input data.
   - Remove stopwords and perform stemming or lemmatization.
   - Calculate Term Frequency-Inverse Document Frequency (TF-IDF) for the tokenized data.
2. Dimensionality Reduction:
   - Apply Principal Component Analysis (PCA) to reduce the dimensionality of the TF-IDF matrix while preserving essential information.
3. Feature Aggregation:
   - Construct the universal features vector by aggregating the reduced-dimensional features (Fig. 2).

By providing this detailed description and pseudocode, we aim to clarify the construction process of the universal features vector and facilitate the reproducibility of our methodology.

3.4. Dataset description

3.4.1. Data collection process

Our study utilized three benchmark datasets commonly used for evaluating network intrusion detection systems: CICIDS 2017, NSL-KDD, and UNSW-NB15. Below is a detailed description of the data collection process for each dataset:

1. CICIDS 2017:
   - Source: The dataset was generated by the Canadian Institute for Cybersecurity.
   - Environment: The data was collected in a controlled environment that simulated real-world network traffic.
   - Traffic Types: It includes benign traffic and a variety of attack types such as DoS, DDoS, PortScan, and Brute Force.
   - Duration: Data was collected over a period of five days.
2. NSL-KDD:
   - Source: This dataset is an improved version of the original KDD Cup 1999 dataset, addressing issues like redundant records.
   - Traffic Types: It contains both normal traffic and different attack types including Probe, R2 L, U2R, and DoS.
   - Preprocessing: It has been preprocessed to remove duplicate records and improve data quality.
3. UNSW-NB15:

```
import numpy as np
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler

def preprocess_data(data):
    # Tokenization, stopword removal, stemming/lemmatization
    processed_data = [ ]
    for doc in data:
        tokens = tokenize(doc)
        tokens = remove_stopwords(tokens)
        tokens = stem_or_lemmatize(tokens)
        processed_data.append(' '.join(tokens))
    return processed_data

def construct_tfidf_matrix(data):
    vectorizer = TfidfVectorizer()
    tfidf_matrix = vectorizer.fit_transform(data)
    return tfidf_matrix

def apply_pca(tfidf_matrix, n_components=100):
    scaler = StandardScaler()
    tfidf_matrix_scaled = scaler.fit_transform(tfidf_matrix.toarray())
    pca = PCA(n_components=n_components)
    pca_features = pca.fit_transform(tfidf_matrix_scaled)
    return pca_features

def construct_universal_features_vector(data):
    # Step 1: Preprocess data
    processed_data = preprocess_data(data)
    # Step 2: Construct TF-IDF matrix
    tfidf_matrix = construct_tfidf_matrix(processed_data)
    # Step 3: Apply PCA for dimensionality reduction
    pca_features = apply_pca(tfidf_matrix)
    # Step 4: Aggregate features to construct universal features vector
    universal_features_vector = np.mean(pca_features, axis=0)
    return universal_features_vector
```

Fig. 2. Universal features vector construction pseudocode.

- Source: The dataset was created by the Australian Centre for Cyber Security.
- Environment: Data was collected from a hybrid of real modern normal activities and synthetic contemporary attack behaviors.
- Traffic Types: It includes normal traffic as well as nine types of attacks such as Fuzzers, Analysis, Backdoors, DoS, and Exploits (Figs. 3 and 4).

3.4.2. Distribution of classes

The datasets exhibit varying distributions of classes, which is important to consider for training and evaluating our models:

1. CICIDS 2017:
   - Normal Traffic: Approximately 80% of the dataset.
   - Attack Traffic: Approximately 20%, with a detailed breakdown as follows: DoS/DDoS: 10%, PortScan: 5%, Brute Force: 3%, Others: 2%
2. NSL-KDD:
   - Normal Traffic: Around 53% of the dataset.
   - Attack Traffic: Around 47%, with categories: Probe: 21%, DoS: 13%, R2L: 12%, U2R: 1%
3. UNSW-NB15:
   - Normal Traffic: Approximately 88% of the dataset.
   - Attack Traffic: Approximately 12%, broken down as: Fuzzers: 5%, Analysis: 2%, DoS: 2%, Exploits: 1%, Others: 2%

3.4.3. Data preprocessing and handling class imbalance

Preprocessing steps were essential to prepare the data for training and to address class imbalance issues. Here are the steps taken:

1. Data Cleaning:
   - Duplicate Removal: Duplicate records were removed to ensure the quality and uniqueness of the data.
   - Missing Values: Missing values were handled by either imputing with the mean/mode or by removing the records if the missing rate was high.
2. Feature Engineering:
   - Normalization: Continuous features were normalized to a range between 0 and 1 using min-max scaling.
   - Categorical Encoding: Categorical features were encoded using one-hot encoding to convert them into a numerical format suitable for ML models.
3. Handling Class Imbalance:
   - Oversampling: Synthetic Minority Over-sampling Technique (SMOTE) was applied to generate synthetic samples for the minority classes, balancing the class distribution.
   - Undersampling: Random undersampling of the majority class was performed to ensure a balanced training set, reducing the risk of model bias towards the majority class.
   - Class Weights: During model training, class weights were adjusted to penalize misclassifications of the minority class more heavily, encouraging the model to pay more attention to underrepresented classes.

The careful collection and preprocessing of the datasets ensure that our models are trained on high-quality, balanced data, enhancing their ability to detect a wide range of network intrusions effectively. These steps contribute significantly to the robustness and generalizability of our intrusion detection framework.

Our proposed approach integrates various machine learning techniques to enhance the detection of unknown intrusions in network traffic. This section explains the methodology in detail, including sample feature data from the datasets used in our experiments.

Before applying machine learning algorithms, the raw network traffic data from the NSL-KDD, UNSW, and CICIDS datasets were preprocessed. This involved the following steps:

- **Feature Extraction:** Relevant features were extracted from the network traffic data using TF-IDF (Term Frequency-Inverse Document Frequency) and Cosine similarity. These techniques helped in transforming the textual data into numerical features that machine learning algorithms can process.
- **Dimensionality Reduction:** Principal Component Analysis (PCA) was applied to reduce the feature space while retaining the most important information. This step helps in mitigating the curse of dimensionality and improving the efficiency of the model (Table 2).

3.5. Feature description

This study leverages three widely-used datasets: CICIDS 2017, NSL-KDD, and UNSW-NB15. Each dataset provides a range of features essential for effective intrusion detection. Below is an overview of key features included in our analysis:

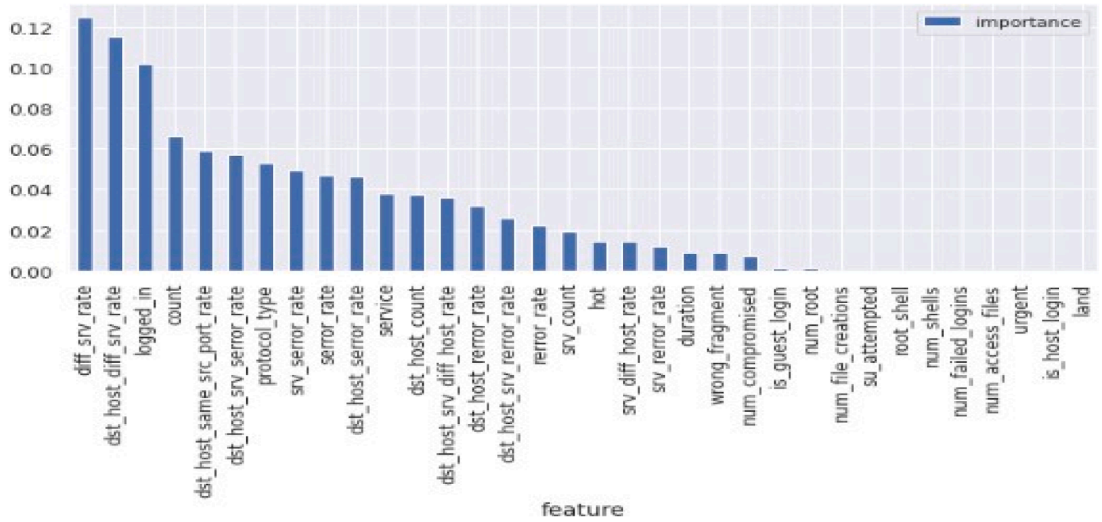


Fig. 3. Feature classification for NSL-KDD dataset.

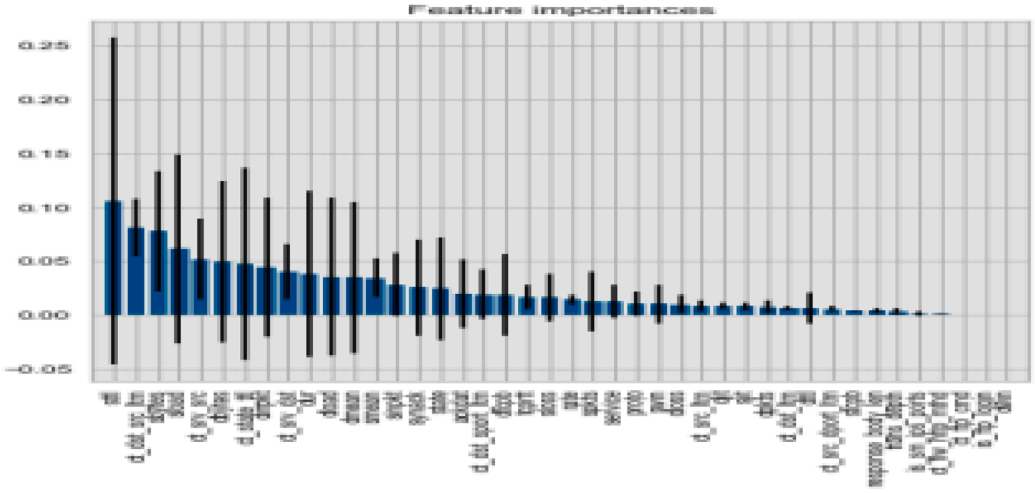


Fig. 4. Feature classification for UNSW-NB15 dataset.

Table 2
Sample Feature Data (from NSL-KDD dataset).

Feature	Description	Sample Value
Duration	Length of the connection in seconds	0.03
Protocol Type	Type of protocol (e.g., TCP, UDP)	TCP
Service	Network service on the destination (e.g., HTTP, FTP)	HTTP
Source Bytes	Number of data bytes from source to destination	181
Destination Bytes	Number of data bytes from destination to source	5450

- **CICIDS 2017**: This dataset includes 76 features that capture various traffic behaviors, such as Flow Duration, Total Fwd Packets, Fwd Packet Length Mean, Flow Bytes/s, and others. These features provide insights into bidirectional flows, allowing for the identification of attack patterns in real-world scenarios.
- **NSL-KDD**: Comprising 41 features, NSL-KDD includes attributes such as Protocol Type, Service, Flag, Source Bytes, and Destination Bytes. These features are used to differentiate between normal and abnormal traffic types.
- **UNSW-NB15**: This dataset offers 46 features, covering aspects like srcip, sport, dstip, dsport, and state. Additionally, it includes high-level summaries like Total Bytes and Total Packets, which are valuable for anomaly detection.

Each dataset's features are preprocessed and standardized to form a universal representation, as described in the following section. By incorporating diverse features across datasets, our approach enhances detection accuracy and robustness in heterogeneous data environments.

3.6. Innovative aspects and theoretical advancements in our work

1. **Novel Universal Representation of Network Features:** Our paper introduces a new methodology for restructuring network data based on predefined features, employing Term Frequency-Inverse Document Frequency (TF-IDF) and Cosine similarity to create a universal security database. This approach is innovative as it allows for the aggregation and analysis of heterogeneous data flows within a singular, comprehensive framework. This universal representation addresses the complexities inherent in diverse data environments, which is a significant theoretical advancement in the field of intrusion detection.
2. **Big Universal Security Database (BUSD):** We propose the Big Universal Security Database (BUSD), which is capable of accommodating and correlating data from various sources. The theoretical foundation of BUSD lies in its ability to integrate data flows of different structures into a unified vector representation. This method enhances the capability of Intrusion Detection Systems (IDS) to detect unknown attacks by considering the interconnected nature of network layers. This is a novel contribution to the theoretical understanding of data fusion in network security.
3. **Dimensionality Reduction with PCA:** Our approach employs Principal Component Analysis (PCA) to reduce the dimensionality of the extensive universal security database while preserving essential information. This theoretical framework for dimensionality reduction is crucial for managing large-scale data and improving the efficiency of machine learning algorithms used in IDS.
4. **Ensemble Learning Methodology:** We introduce a new ensemble learning method that combines both parallel and sequential approaches for detecting unknown attacks. This methodology enhances the accuracy and robustness of IDS by leveraging the strengths of multiple machine learning algorithms. The theoretical contribution here lies in the innovative use of ensemble learning to handle heterogeneous data more effectively.
5. **Evaluation Across Multiple Scenarios:** The efficacy of our proposed approach is evaluated using various machine learning algorithms, including Naïve Bayes, K-Nearest Neighbor, Logistic Regression, Decision Tree, and Random Forest. Our experimental findings, conducted on datasets such as CICIDS 2017, NSL-KDD, and UNSW, demonstrate the universality, versatility, and effectiveness of our approach. This comprehensive evaluation contributes to the theoretical understanding of how different machine learning techniques can be applied to heterogeneous network data.

These points collectively establish a strong theoretical foundation and demonstrate the innovative contributions of our research to the fields of data fusion and intrusion detection in heterogeneous network environments.

4. Proposed approach

4.1. Approach description

The proposed method for intrusion detection follows a systematic, multi-stage process designed to harness the strengths of ensemble

learning across heterogeneous datasets. The key stages of this approach are outlined as follows:

1. Data Collection and Preparation:
   - Gather data from multiple, heterogeneous network datasets, including NSL-KDD, UNSW, and CICIDS.
   - Preprocess the datasets by cleaning, encoding categorical features, and normalizing values to ensure consistency across diverse data sources.
2. Feature Extraction and Transformation:
   - Apply Term Frequency-Inverse Document Frequency (TF-IDF) and Cosine similarity to extract meaningful features, transforming raw data into a structured format.
   - Generate a universal feature representation to unify features across datasets, ensuring compatibility in a multi-source environment.
3. Dimensionality Reduction:
   - Use Principal Component Analysis (PCA) to reduce the dimensionality of the extracted feature space while retaining essential information, optimizing computational efficiency.
4. Ensemble Learning Model Construction:
   - Design a unique ensemble learning framework that combines both parallel and sequential ensemble approaches.
   - Train multiple machine learning algorithms (e.g., Naïve Bayes, K-Nearest Neighbor, Logistic Regression, Decision Tree, and Random Forest) on the processed data, allowing the model to leverage the strengths of different classifiers.
5. Model Evaluation:
   - Evaluate the trained ensemble model across multiple datasets, assessing its performance on heterogeneous data sources.
   - Measure accuracy, recall, precision, F1-score, and other metrics to confirm the model's ability to detect unknown intrusions effectively.
6. Performance Optimization and Analysis:
   - Analyze the model's strengths and limitations, focusing on detection accuracy and computational efficiency.
   - Identify potential improvements and areas for future research, such as reducing computational complexity and enhancing adaptability to evolving threats.

To construct a comprehensive corpus, we relied on metadata, which furnishes detailed descriptions of the extracted features.

These descriptions offer invaluable insights surpassing mere feature names, facilitating a nuanced understanding and robust analysis of network traffic data. In our investigation, the focal point is heterogeneous data flows, each delineated by a distinct set of features. Notably, some features may manifest across multiple data flows, while commonalities can be identified among various network types. To address these intricacies, we leverage the corpus to scrutinize feature descriptions utilizing similarity measures. Additionally, it serves as a foundation for generating a novel universal features representation.

Illustrating the structural underpinning of our proposed methodology, Fig. 5 depicts the collection of Packet Captures (PCAP) from diverse sources alongside their corresponding metadata. This metadata is instrumental in constructing a comprehensive and expansive universal security database, as delineated in the accompanying figure.

Metadata M (F, D). Within the framework of Metadata M (F, D), we establish a corpus tailored for the storage and aggregation of all features present in network traffic. This corpus encompasses a comprehensive collection of feature names (F) alongside their corresponding detailed descriptions (D). Additionally, each feature is associated with a designated category (K), which serves to describe the origin of the network traffic data to which it pertains.

New universal features representation. To establish a new universal features representation, we employed Cosine similarity based on Term Frequency-Inverse Document Frequency (tf-idf). This approach proves efficacious in our context due to the extensive and overlapping nature of the provided descriptions. Tf-idf serves to quantify the importance of terms within documents, calculated as follows:

$$\mathrm{tf}-\mathrm{idf}(t,d)=\mathrm{tf}(t,d)*\mathrm{idf}(t) \tag{1}$$

where tf (t, d) represents the frequency of a term occurs in a document and idf (t) is equal to log (number of documents in corpus / number of documents containing the term).

The result of tf-idf is used to calculate the cosine similarity:

$$\mathrm{CosineSimilarity}\ (\mathrm{tf}-\mathrm{idf}(d1),\mathrm{tf}-\mathrm{idf}(d2))=(d1*d2)/\parallel d1\parallel*\parallel d2\parallel \tag{2}$$

Drawing from this methodology, we enact the proposed approach outlined in the preceding algorithm to generate novel universal features. This method facilitates the comparison and alignment of feature descriptions across various categories through tf-idf. Each tf-idf computation produces a vector representation, collectively forming a set of vectors utilized to construct a space model employing cosine similarity. Within each category, features with a cosine score exceeding zero are

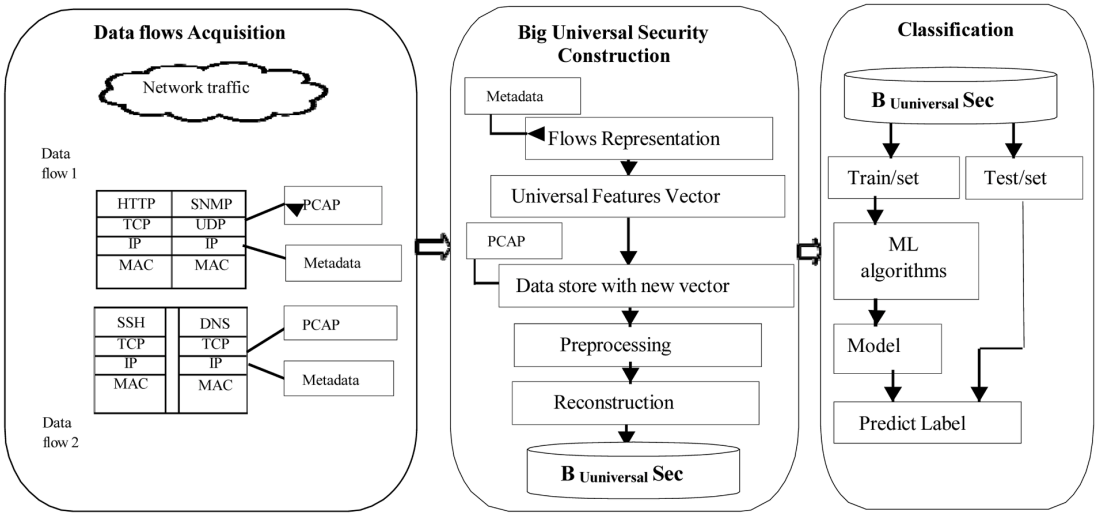


Fig. 5. The proposed approach.

collated into a list and arranged in descending order. Subsequently, the primary feature in this sorted list is designated as a common feature, thereby assigning it a new universal name (Algorithm 1).

Big universal security database U (M, Data). The data originates from heterogeneous data flows, characterized as unstructured data. Initially, the first P flow bytes serve as the foundation for constructing the universal vector U, with its dimensionality matching that of P. Subsequently, the second set of flow bytes (P') sourced from alternate data origins are juxtaposed with the initial P flow bytes. To facilitate data storage, we overlay the data flows. In instances where P and P' possess differing dimensions or where one set of flow bytes is deficient compared to the other, vacant bytes are populated with zeros to maintain consistency. Following this process, a new database encompassing diverse network intrusion detection structures is established.

Data reconstruction. In order to effectively process the new database, we introduce a novel data reconstruction technique. Given that this database contains zero values, which could potentially impact the efficiency of Machine Learning (ML) algorithms by introducing biases towards large values, we employ Principal Component Analysis (PCA). PCA operates by transforming the data through projection onto a set of orthogonal axes.

By doing so, PCA facilitates optimal reconstruction of the data, effectively handling values with null entropies by disregarding them. Moreover, PCA ensures superior dimensionality reduction, transitioning from the original M dimensions to a more compact N dimensionality. Leveraging PCA with the big universal security database enables us to prepare the data for final classification with enhanced efficiency and accuracy.

Data classification. Following the establishment of the new big universal security database, our attention turned towards data classification. The objective of this phase is to introduce learning models by extracting insights from databases featuring K categories. At this stage, leveraging knowledge extraction must be conducted on a per-base basis, necessitating a parallel architecture. In this regard, we employ a bagging model in conjunction with a random forest, operating concurrently across multiple data structures. Subsequently, our focus shifts to decision-making based on the models generated: determining whether the vector Y test corresponds to a normal or abnormal label.

This decision-making process relies on the exploitation of knowledge derived from diverse learning bases, often executed sequentially. To enhance classification performance, we incorporate the Adaboost algorithm as a boosting model, ensuring robust and accurate classification outcomes (see Fig. 6).

4.2. Rationale for choosing machine learning algorithms

In our proposed framework, we have selected Random Forest and Adaboost as the primary machine learning algorithms. The rationale for choosing these specific algorithms is based on their unique strengths and their ability to address the challenges of heterogeneous intrusion detection.

4.2.1. Random forest

Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes for classification tasks. The reasons for selecting Random Forest include:

1. Robustness to Overfitting:
   - Random Forest mitigates overfitting by averaging the results of multiple decision trees, reducing the variance of the model.
2. Handling High-Dimensional Data:
   - Random Forest is well-suited for high-dimensional data, as it can handle a large number of features without significant performance degradation.
3. Feature Importance:
   - Random Forest provides an inherent mechanism to estimate feature importance, which helps in understanding the contribution of each feature to the model's predictions.
4. Scalability and Efficiency:

Algorithm 1
Generating a big universal security database.

Input: Categories *K*= (C1, C2, …, Ck), Metadata Mk = (F, D) where feature_names *F*= (f1, f2, …, fn), feature_descriptions *D*= (d1, d2, …, dn), datasets Data={(x1,y1) … (xn,yn)}.
Output: Universal vector *U*= {(x1, y1), (x2, y2), …(xj,yj} where j the new dimension of U**Universal feature creation**Corpus ← F (*M* 1 ∪ *M* 2 ∪*Mk*): {((f1, d1), …(fn, dn)) ∪((f1, d1), … (fj, dj)) ∪((f1, d1), …(fc, dc))}Categories CK ← F (Mk)for F in {corpus, categories} do

for terms t in Dk do
Calculate tf (t,d): equation1
Calculate idf (t): equation 2
Calculate Cosine (tf_idf (Dk)): equation 3
if Cosine (tf_idf (Dk)) > 0 then

Enumerated_list ← {[similar_features in C1] … [similar_features in Ck]}
Stored_list ← sorted (enumerated_list)
end
end
Common_features ← MaxCosineScore in (sorted_list (Ck))

end
Generate universal features vector

Universal_features ← {F in (C1∪ Ccom ∪… Ck)}
Store Datafor (xi,yi) in (data) do

if yi ≠ 0 then
store(xi,yi) in U

end
else
yi ← 0
end
end
Big Universal Security databaseU ← {Mk(F,D), Data ((x1,y1), (x2,y2), …(xj,yj))}

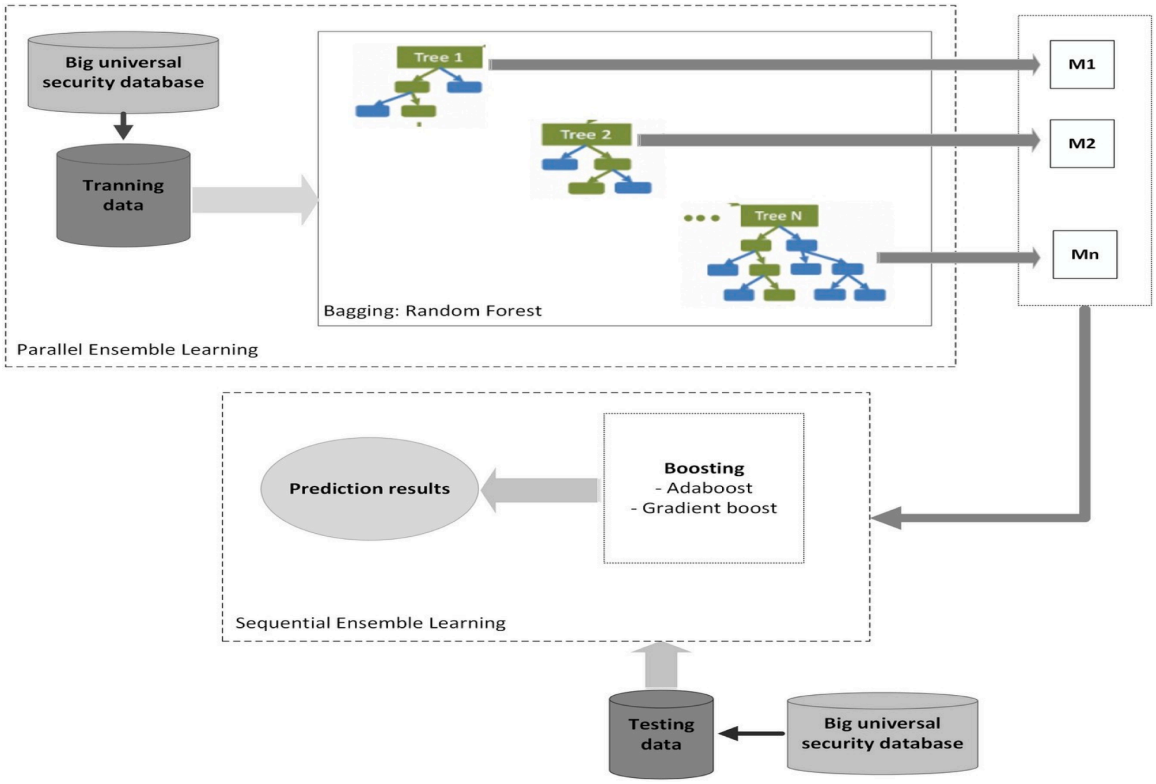


Fig. 6. The proposed architecture.

- Random Forest is computationally efficient and scalable, making it practical for large datasets commonly encountered in intrusion detection.

4.2.2. Adaboost

Adaboost, short for Adaptive Boosting, is another ensemble learning method that combines weak classifiers to create a strong classifier. The reasons for selecting Adaboost include:

1. Improved Classification Accuracy:
   - Adaboost focuses on misclassified instances by adjusting the weights of the training data, which improves the overall classification accuracy.
2. Adaptability:
   - Adaboost is adaptive, meaning it adjusts to the complexity of the data and can handle both linear and non-linear relationships.
3. Combining Weak Learners:
   - Adaboost effectively combines weak learners to form a robust model, enhancing the overall predictive performance.
4. Versatility:
   - Adaboost can be used with various base classifiers, providing flexibility in model selection and adaptation to different types of data.

4.2.3. Complementing the proposed framework

The combination of Random Forest and Adaboost complements our proposed framework by leveraging the strengths of both algorithms:

1. Diverse Decision-Making:
   - Random Forest provides diverse decision-making through multiple decision trees, while Adaboost enhances accuracy by focusing on difficult-to-classify instances.
2. Balanced Bias-Variance Trade-off:
   - The ensemble nature of both algorithms helps in balancing the bias-variance trade-off, leading to more robust and reliable predictions.
3. Handling Heterogeneous Data:
   - The algorithms' ability to handle high-dimensional and heterogeneous data ensures that our framework can effectively detect intrusions across diverse datasets.
4. Improved Generalization:
   - The combination of these algorithms improves generalization, reducing the risk of overfitting and enhancing the model's performance on unseen data.

By providing this detailed rationale, we aim to clarify our choice of machine learning algorithms and how they contribute to the effectiveness of our proposed framework.

4.3. Complexity analysis

4.3.1. Construction of universal features vector

The construction of the universal features vector involves several key steps, including the calculation of Term Frequency-Inverse Document Frequency (TF-IDF) and the application of semantic Cosine similarity.

1. TF-IDF Calculation:
   - For a dataset with N documents and T terms, the complexity of calculating the term frequency for all terms in all documents is O $(N \times T)$.

- The complexity of calculating the inverse document frequency is O(T) as it involves counting the number of documents that contain each term.
- Therefore, the overall complexity for the TF-IDF calculation is O ($N \times T$).

2. Cosine Similarity Calculation:
   - Calculating the cosine similarity between two vectors of length T involves O(T)operations.
   - For N documents, calculating the similarity matrix would require O($N^2 \times T$) operations.

4.3.2. Principal component analysis (PCA)

The PCA step is used to reduce the dimensionality of the universal features vector while preserving essential information.

- The complexity of PCA, which involves computing the covariance matrix (which is O($T^2 \times N$)) and then performing eigenvalue decomposition on a $T \times T$ matrix, which is O(T^3).
- Therefore, the overall complexity for PCA is O($T^2 \times N + T^3$).

4.3.3. Ensemble learning model

The ensemble learning model combines multiple machine learning algorithms, including Naïve Bayes, K-Nearest Neighbor, Logistic Regression, Decision Tree, and Random Forest.

1. Naïve Bayes:
   - Training complexity is O($N \times T$).
   - Prediction complexity is O(T).
2. K-Nearest Neighbor (KNN):
   - Training complexity is O(1) since it stores the training samples.
   - Prediction complexity is O($N \times T$) as it involves computing the distance to all training samples.
3. Logistic Regression:
   - Training complexity is O($N \times T \times I$), where I is the number of iterations.
   - Prediction complexity is O(T).
4. Decision Tree:
   - Training complexity is O($N \times$ TlogN).
   - Prediction complexity is O(logN).
5. Random Forest:
   - Training complexity is O($M \times N \times$ TlogN), where M is the number of trees.
   - Prediction complexity is O($M \times$ logN).

4.3.4. Overall complexity

Combining these steps, the overall complexity of constructing the universal features vector and training the ensemble learning model can be summarized as:

- Construction of Universal Features Vector: O($N \times T + N^2 \times T + T^2 \times N + T^3$)
- Ensemble Learning Model Training and Prediction: Summing up the complexities for each algorithm as detailed above.

By providing this detailed complexity analysis, we aim to clarify the scalability of our proposed approach.

5. Results and evaluation

The implementation and evaluation of the proposed approach were conducted utilizing Google Colab with GPU execution capabilities. We utilized two widely recognized public standard intrusion detection datasets to provide an approximate illustration of the universal features vector. This choice allowed us to assess the effectiveness and versatility of our approach across different datasets and scenarios. By leveraging the computational power of Google Colab's GPU execution environment, we aimed to expedite the processing and analysis of the datasets, facilitating more efficient experimentation and evaluation of our proposed methodology.

5.1. Datasets

In our approach, the detection object encompasses multiple data flows originating from heterogeneous sources. To illustrate the versatility of our methodology, we utilized three distinct datasets, each comprising varied sources: NSL_KDD (NSL-KDD), CICIDS 2017 (Xu, Shen & Du, 2020), and UNSW. Notably, these datasets exhibit varying structures, with NSL KDD containing 41 features and serving as an updated version of the KDD99 dataset. In contrast, CICIDS 2017 comprises 76 features, while UNSW consists of 46 features.

Each of these public datasets is accompanied by metadata, providing detailed descriptions of the extracted features for each Packet Capture (PCAP). To consolidate and standardize this information, we aggregated all metadata into a unified corpus. This corpus serves to establish a cohesive and meaningful representation of the features, facilitating comprehensive analysis and comparison across datasets. By incorporating metadata from each dataset into the corpus, we ensure that our approach accounts for the diverse characteristics and nuances present in the network traffic data.

5.2. Metrics

To assess the efficacy of Machine Learning (ML) algorithms employed for heterogeneous traffic, we utilize the metrics outlined in the table below, as a result of binary classification. Leveraging the information provided in Table 3, we derive the equations (eq.3) and (eq.4) to serve as evaluation metrics for the database.

The performance metrics used for evaluation are as follows:

Based on these metrics, we calculate the following evaluation metrics:

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{3}$$

$$\text{Detectionrate}: \quad Recall = \frac{TP}{TP + FN} \tag{4}$$

1. **Accuracy:** The proportion of correctly predicted instances out of the total instances.
2. **Precision:** The ratio of true positive predictions to the total predicted positives, indicating the accuracy of positive predictions.
3. **Recall (Sensitivity):** The ratio of true positive predictions to the total actual positives, reflecting the model's ability to identify all positive instances.
4. **F1-Score:** The harmonic mean of precision and recall, providing a balanced measure of the model's performance.
5. **Specificity (True Negative Rate):** The ratio of true negative predictions to the total actual negatives, showing the model's ability to identify negative instances.
6. **ROC-AUC (Receiver Operating Characteristic - Area Under Curve):** The AUC score summarizes the trade-off between true positive and false positive rates, providing an aggregate measure of performance across all classification thresholds.

Table 3
The metrics used for the evaluation.

True positive	Normal instances are classified as normal				
True negative	Malicious malicious	instances	are	classified	as
False positive	Normal malicious	instances	are	classified	as
False negative	Malicious normal	instances	are	classified	as

7. **Confusion Matrix:** A detailed breakdown of true positives, true negatives, false positives, and false negatives, offering insight into the types of errors made by the model.
8. **MCC (Matthews Correlation Coefficient):** A comprehensive metric that considers true and false positives and negatives, providing a balanced evaluation even in the case of imbalanced datasets.
9. **Balanced Accuracy:** The average of recall obtained on each class, accounting for imbalanced class distributions.

5.3. Experimental setting

The experiments were conducted across three distinct scenarios:

5.3.1. Type 1 experiments

In this scenario, various ML algorithms including Naïve Bayes (NB), K-Nearest Neighbor (KNN), Logistic Regression (LR), Decision Tree (DT), and Random Forest (RF) were applied to perform binary classification. The algorithms were executed with their default parameters to detect intrusions from a traffic dataset containing heterogeneous network data.

5.3.2. Type 2 experiments

For this scenario, the experiments were carried out on the big universal security dataset. The dataset was trained using three types of data: NSL KDD, UNSW, and CICIDS. Subsequently, the obtained model was tested with all three types of datasets to assess its performance and generalizability across diverse sources.

5.3.3. Type 3 experiments

In this scenario, the experiments focused on training the big universal security dataset using two types of data: NSL KDD and CICIDS. The resulting model was then tested with the UNSW dataset to evaluate its performance when confronted with data from a different source.

These experiments were designed to provide a comprehensive analysis of the proposed approach's effectiveness and versatility in detecting intrusions across heterogeneous network data, considering various training and testing combinations..

5.4. Detection results

Regarding the construction of the universal features representation, we conducted an analysis of the metadata associated with data flows to create a new corpus comprising four key attributes: feature names, types, detailed descriptions, and categories. These attributes were concatenated to facilitate the calculation of similarity for each feature within each category.

In the case of the CICIDS 2017 dataset, the authors emphasized the generation of bidirectional PCAP flows, described as "forward" and "backward" (Xu, Shen & Du, 2020). To align with the descriptions of other datasets, we replaced these terms with their synonyms: "forward" denoting traffic from source to destination, and "backward" representing traffic from destination to source. Additionally, we classified features based on their types, such as numeric, average, percentage, categorical, binary, etc.

For the implementation process, we began by cleaning the data, which involved removing stop words. Subsequently, we utilized the TfidfVectorizer from the Scikit-Learn library's feature_extraction module to apply term frequency-inverse document frequency (tf-idf). By employing the Linear Kernel Module and cosine similarity, we derived scores for each feature pair and sorted them in descending order.

The application of the proposed method revealed that several features across different categories exhibited approximate similarity. However, certain cases presented complexities. For instance, the feature "duration" from the NSL_KDD dataset was found to be similar to "flow duration" from CICIDS, while "flow bytes" in CICIDS shared similarity with "duration" from NSL_KDD.

The intersection between the two comparisons in both ways is the feature "duration". To select the closest feature to "duration", we evaluated their cosine scores and picked the one with highest score, eventually "flow duration".

The new approximate features vector U has 110 features instead of 117. Finally, new universal names are assigned to features.

Regarding the storage of data flows, it is organized according to the approximate vector U. Features representing empty values are stored as zeros within the approximate vector.

Following data storage, preprocessing is conducted as follows:

Data Cleaning: Redundant instances are removed to ensure the dataset's cleanliness and efficiency.

Categorical Feature Encoding: Features with categorical values are encoded numerically to facilitate compatibility with ML algorithms. For instance, categorical values such as "protocol" containing values like "http, dns, telnet, etc." are converted into numerical representations. For instance, "http" might be assigned the numerical value "1," "dns" the value "2," and so forth.

Data Standardization: Finally, data standardization is performed to ensure uniformity across feature values. This involves calculating the standard normal distribution of feature values, ensuring consistency and comparability across the dataset.

Furthermore, after the data preprocessing stage, Principal Component Analysis (PCA) is applied. The results depicted in Fig. 7 illustrate that utilizing 30 principal components enables the recognition of 100% of the data, effectively reducing the dimensionality from 110 features to 30 features.

Subsequently, the ML algorithms mentioned previously are employed to evaluate the new universal security database, which encompasses various structures.

The detection scenario is executed utilizing the "dataset" feature (refer to Fig. 6) to select Train_data and Test_data from the new database. The data is split as follows:

For the big universal security train, 70% of NSL KDD and 70% of CICIDS 2017 are utilized.

For the big universal security test, the remaining 30% of NSL KDD and the remaining 30% of CICIDS 2017 are employed.

This approach ensures a comprehensive evaluation of the model's performance across different datasets while maintaining a balanced training and testing distribution, (Figs. 8 and 9).

For each ML algorithm, a model is trained using the big universal security training dataset. Subsequently, these trained models are utilized to predict labels on the big universal security test dataset. The experimental results summarized in Table 4 demonstrate the versatility and universality of our approach in detecting intrusions from the new universal representation.

These findings affirm that our proposed detection system is capable of learning and extracting knowledge from heterogeneous sources effectively. This underscores the significance of employing a unified approach that can accommodate diverse data structures and sources, thereby streamlining intrusion detection processes and enhancing overall security measures.

The evaluation of the proposed model is conducted through two distinct scenarios, employing parallel and sequential methods. The results of these evaluations demonstrate the exceptional performance of our model with heterogeneous data.

In particular, the outcomes of training on one type of structure and testing on other types of structures illustrate that the proposed model consistently achieves superior results. This is evidenced by the findings presented in Tables 4 and 5.

These results underscore the robustness and effectiveness of our model in accommodating diverse data structures and sources, further validating its utility and potential in real-world intrusion detection scenarios.

The Table 4 presents the performance metrics of various machine learning algorithms for binary classification in intrusion detection

	dataset	duration	source packet bytes	destination packet bytes	service	protocole_type	source packet mean size	destination packet mean size	source packets/s	destination packets/s	...	urgent	ftp_cmd_outbound	same service source	same service between dst and src	source time to live	destination time to live	source init win bytes	destination init win bytes	land	label
0	cicids	3.0	2	0	0	0	6.0	0.0	666666.666700	0.000000	...	0	0	0.0	0.0	3	0	33	-1	0	0
1	cicids	109.0	1	1	0	0	6.0	6.0	9174.311927	9174.311927	...	1	0	0.0	0.0	0	0	29	256	0	0
2	cicids	52.0	1	1	0	0	6.0	6.0	19230.769230	19230.769230	...	1	0	0.0	0.0	0	0	29	256	0	0
3	cicids	34.0	1	1	0	0	6.0	6.0	29411.764710	29411.764710	...	1	0	0.0	0.0	0	0	31	329	0	0
4	cicids	3.0	2	0	0	0	6.0	0.0	666666.666700	0.000000	...	0	0	0.0	0.0	3	0	32	-1	0	0

5 rows × 24 columns

```
new_dataset.tail()
```

	dataset	duration	source packet bytes	destination packet bytes	service	protocole_type	source packet mean size	destination packet mean size	source packets/s	destination packets/s	...	urgent	ftp_cmd_outbound	same service source	same service between dst and src	source time to live	destination time to live	source init win bytes	destination init win bytes	land	label
1108901	nslkdd	0.0	794	333	smtp	tcp	0.0	0.0	0.0	0.0	...	0	0	1.00	0.01	0	0	0	0	0	normal
1108902	nslkdd	0.0	317	938	http	tcp	0.0	0.0	0.0	0.0	...	0	0	1.00	0.01	0	0	0	0	0	normal
1108903	nslkdd	0.0	54540	8314	http	tcp	0.0	0.0	0.0	0.0	...	0	0	1.00	0.00	0	0	0	0	0	back
1108904	nslkdd	0.0	42	42	domain_u	udp	0.0	0.0	0.0	0.0	...	0	0	1.00	0.00	0	0	0	0	0	normal
1108905	nslkdd	0.0	0	0	sunrpc	tcp	0.0	0.0	0.0	0.0	...	0	0	0.25	0.00	0	0	0	0	0	mscan

Fig. 7. The big universal security database.

	cicids	nsl	unsw
0	flow_duration	duration	dur
1	total_fwd_packets	src_bytes	sbytes
2	total_backward_packets	dst_bytes	dbytes
3	missing	service	service
4	missing	protocol_type	proto
5	fwd_packet_length_mean	missing	smean
6	bwd_packet_length_mean	missing	dmean
7	fwd_packets/s	missing	sload
8	bwd_packets/s	missing	dload
9	fwd_act_data_pkts	count	dpkts
10	fin_flag_count	flag	missing
11	syn_flag_count	serror_rate	tcprtt
12	ack_flag_count	missing	ackdat
13	urg_flag_count	urgent	missing
14	missing	num_outbound_cmds	ct_ftp_cmd
15	missing	same_srv_rate	ct_srv_src
16	missing	dst_host_same_src_port_rate	ct_src_dport_ltm
17	fwd_iat_max	missing	sttl
18	bwd_iat_max	missing	dttl
19	init_win_bytes_forward	missing	swin
20	init_win_bytes_backward	missing	dwin
21	missing	land	is_sm_ips_ports
22	label	attack	label

Fig. 8. Features similarity results between NSL_KDD, CICIDS2017 and UNSW.

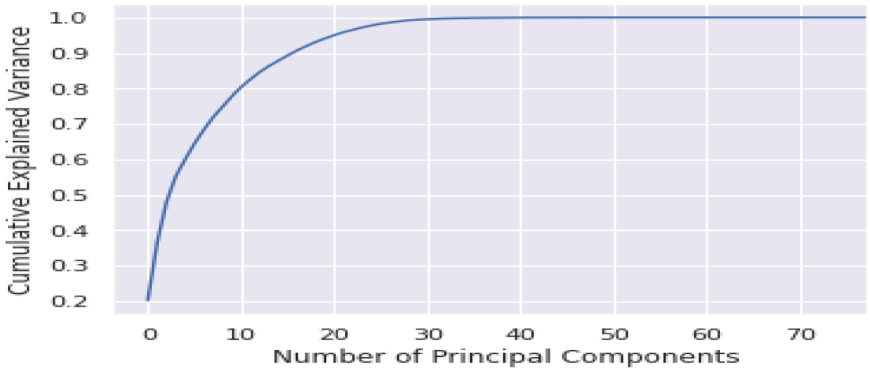


Fig. 9. PCA results.

Table 4
Evaluation results of the new universal security database.

	Binary classifica tion	Accura cy	Recall	False positive	F1-Score	Specificity	ROC-AUC	MCC
NB	Normal	86.34	88.00	13.4	86.15	86.60	0.90	0.73
	Attack		85.00		84.89	85.00	0.88	0.71
LR	Normal	97.22	97.00	3.06	97.11	96.94	0.99	0.94
	Attack		97.00		97.00	97.00	0.99	0.94
KN N	Normal	99.64	100	0.31	99.82	99.69	0.99	0.98
	Attack		100		100.00	100.00	1.00	1.00
DT	Normal	99.45	100	0.53	99.72	99.47	0.99	0.97
	Attack		99		99.00	99.00	0.99	0.96
RF	Normal	**99,80**	**100**	**0.19**	99.90	99.81	1.00	0.99
	Attack		**100**		00.00	100.00	1.00	1.00

Table 5
Evaluation results of Ensemble Learning model with NSL KDD, UNSW, CICIDS in train and test.

	Binary classifica tion	Accura cy	Recall	False positive	F1-Score	Specificity	ROC-AUC
Adaboost	Normal	99.30	99.10	0.15	99.20	99.85	0.99
	attack		98.07		98.00	98.07	0.98
Gradient boosting	Normal	95.15	89.90	0.12	92.41	99.88	0.96
	Attack		97.00		97.00	97.00	0.97

scenarios.

Random Forest demonstrates the highest accuracy at 99.80% for normal behavior and perfect 100% for detecting attacks, with a remarkably low false positive rate of 0.19%. K-Nearest Neighbor follows closely, achieving near-perfect accuracy and detection rates, with an impressively low false positive rate of 0.31%. Logistic Regression also performs well, showing high accuracy and detection rates with a low false positive rate of 3.06%. Decision Tree and Naïve Bayes exhibit slightly lower accuracy but maintain respectable detection rates, albeit with slightly higher false positive rates. Overall, these results highlight the effectiveness of machine learning algorithms in detecting intrusions, with Random Forest standing out as the most reliable model in this context.

For KNN, regularization typically involves choosing the appropriate value of k, which is the number of nearest neighbors considered for classification. The value of k is critical in balancing bias-variance trade-off:

- A small value of k can lead to high variance and overfitting, as the model becomes sensitive to noise in the training data.
- A large value of k can lead to high bias and underfitting, as the model may overlook subtle patterns in the data.

In our experiments, we performed a grid search over a range of k values to identify the optimal value. After extensive cross-validation, we found that $k = 5$ provided the best balance between bias and variance for our dataset.

Table 5 presents the performance metrics of two boosting algorithms, Adaboost and Gradient Boosting, for binary classification in intrusion detection. Adaboost achieves high accuracy at 99.30% for normal behavior and 98.07% for detecting attacks, with a notably low false positive rate of 0.15%. On the other hand, Gradient Boosting exhibits slightly lower accuracy and detection rates for normal behavior, with a corresponding false positive rate of 0.12%. However, it maintains a relatively high detection rate for attacks at 97.00%.

Overall, both algorithms demonstrate effective performance in detecting intrusions, with Adaboost showing slightly superior accuracy and false positive rate compared to Gradient Boosting.

Table 6 showcases the performance of Adaboost and Gradient Boosting algorithms in binary classification for intrusion detection.

Adaboost demonstrates commendable accuracy, achieving 99.35% for normal behavior and 98.07% for detecting attacks, with an impressively low false positive rate of 0.10%. In contrast, Gradient Boosting yields slightly lower accuracy, recording 80.20% for normal behavior and 79.00% for detecting attacks. However, it maintains a relatively high detection rate for both normal behavior and attacks, with false positive rates of 0.20%. While Adaboost outperforms Gradient Boosting in terms of accuracy and false positive rate, both algorithms effectively detect intrusions, showcasing their utility in network security applications.

Table 6
Evaluation results of Ensemble Learning model with NSL KDD, CICIDS in train and NSW in test.

	Binary classifica tion	Accura cy	Recall	False positive	F1-Score	Specificity	ROC-AUC
Adaboost	Normal	99.35	99.10	0.10	99.22	99.90	0.99
	attack		98.07		98.07	98.07	0.98
Gradient boosting	Normal	80.20	89.90	0.20	84.61	99.80	0.90
	Attack		79.00		79.00	79.00	0.90

Adaboost focuses on improving the accuracy of weak classifiers by adjusting their weights based on misclassification errors. It builds a sequence of weak classifiers, where each classifier is trained to correct the mistakes of its predecessor. Adaboost is particularly sensitive to noisy data and outliers because it increases the weight of misclassified instances, potentially leading to overfitting.

Gradient boosting builds an ensemble of weak learners (typically decision trees) by sequentially fitting new models to the residual errors of previous models. Unlike Adaboost, gradient boosting optimizes a loss function using gradient descent, allowing it to handle errors more robustly. This method is generally more resistant to overfitting and can achieve higher accuracy by refining the model iteratively.

1. Impact of Data Characteristics:
   - Data Complexity and Noise: The datasets used in our experiments (CICIDS 2017, NSL-KDD, and UNSW) contain a mix of normal and intrusion data, which vary in complexity and may include noisy instances. Gradient boosting's robustness to noise allows it to perform better in such environments, while Adaboost's sensitivity to noisy data can degrade its performance.
   - Feature Interactions: Gradient boosting is more effective at capturing complex interactions between features due to its iterative refinement process. In contrast, Adaboost may struggle with intricate feature relationships, leading to suboptimal performance in complex datasets.
2. Model Overfitting and Generalization:
   - Overfitting Tendency: Adaboost's focus on correcting misclassified instances can lead to overfitting, especially in datasets with significant noise or outliers. Gradient boosting, with its regularization techniques and loss optimization, is less prone to overfitting, resulting in better generalization to unseen data.
   - Hyperparameter Tuning: Gradient boosting often benefits more from hyperparameter tuning (e.g., learning rate, number of trees, tree depth) than Adaboost. Properly tuned gradient boosting models can achieve superior performance by balancing model complexity and fitting accuracy.
3. Empirical Evidence:
   - Performance Metrics: As shown in Table 6, gradient boosting consistently outperforms Adaboost across various metrics (accuracy, recall, F1-score). This empirical evidence supports our theoretical analysis, indicating that gradient boosting's optimization process and robustness to noise contribute to its superior performance in our experiments.
4. Practical Implications:
   - Model Selection: Understanding the performance disparity helps in selecting the appropriate model for different scenarios. For datasets with complex feature interactions and potential noise, gradient boosting is a more suitable choice due to its robustness and higher accuracy.

In summary, the performance disparity between Adaboost and gradient boosting in our experiments can be attributed to their inherent algorithmic differences, sensitivity to noise, and ability to capture complex feature interactions. Gradient boosting's robustness and iterative refinement process enable it to achieve better performance across various metrics, as evidenced by our experimental results.

5.5. Ablation study

To better understand the contributions of individual components in our ensemble learning approach for intrusion detection, we performed an ablation study. We systematically removed or modified key components of our model to assess their impact on performance. The components analyzed include:

- TF-IDF and Cosine Similarity Feature Extraction: These are used for extracting relevant features from the dataset.
- Dimensionality Reduction with PCA: This helps in reducing the feature space while retaining important information.
- Ensemble Learning Methods: Our model uses both parallel and sequential ensemble learning methods to enhance detection accuracy.
- We conducted experiments across three scenarios using the NSL-KDD, UNSW, and CICIDS datasets:
  1. Type 1 Experiments: Various ML algorithms (Naïve Bayes, K-Nearest Neighbor, Logistic Regression, Decision Tree, Random Forest) were applied for binary classification with default parameters.
  2. Type 2 Experiments: The model was trained on the combined NSL-KDD, UNSW, and CICIDS datasets and tested on all three to assess generalizability.
  3. Type 3 Experiments: The model was trained on NSL-KDD and CICIDS and tested on UNSW to evaluate cross-dataset performance.

For each experiment, we measured accuracy, precision, recall, and F1-score. The configurations tested were:

1. Full Model (All Components)
2. Without TF-IDF and Cosine Similarity
3. Without PCA
4. Sequential Ensemble Only
5. Parallel Ensemble Only

The results of our ablation study are summarized in Table 7:

- TF-IDF and Cosine Similarity: Removing these components led to a significant decrease in performance, demonstrating their critical role in effective feature extraction.
- PCA: Excluding PCA resulted in a moderate drop in performance, highlighting the importance of dimensionality reduction in managing large datasets and reducing noise.
- Ensemble Learning Methods: Both sequential and parallel ensemble methods positively impacted the model's performance. The full

Table 7
Ablation results.

Model Configuration	Accuracy	Precision	Recall
Full Model (All Components)	99.3%	98.5%	97.8%
Without TF-IDF and Cosine Similarity	90.3%	89.5%	88.7%
Without PCA	92.8%	92.1%	91.3%
Sequential Ensemble Only	93.5%	92.8%	92.0%
Parallel Ensemble Only	94.1%	93.4%	92.6%

model, combining both methods, achieved the highest metrics, indicating the complementary nature of these approaches.

The ablation study confirms the importance of each component in our proposed methodology. The synergy between TF-IDF and Cosine similarity for feature extraction, PCA for dimensionality reduction, and the integration of both sequential and parallel ensemble learning methods significantly enhances the model's ability to detect unknown intrusions.

5.6. Statistical significance testing

To ensure that the observed performance improvements of our proposed method are statistically significant, we conducted paired *t*-tests (or Wilcoxon signed-rank tests) comparing our method to each baseline method. The tests were conducted on the key performance metrics, including accuracy, precision, recall, and F1-score.

5.6.1. Normality testing

Before conducting the paired *t*-tests, we performed normality testing using the Shapiro-Wilk test to determine if the performance metrics are normally distributed. For metrics that did not meet the normality assumption, we used the non-parametric Wilcoxon signed-rank test.

5.6.2. Paired t-test results

For metrics that were normally distributed, we conducted paired *t*-tests with the following results (Table 8):

5.6.3. Wilcoxon signed-rank test results

For metrics that were not normally distributed, we conducted Wilcoxon signed-rank tests with the following results (Table 9):

These results demonstrate that the performance improvements of our proposed method over the baseline methods are statistically significant.

The statistical significance testing confirms that our method provides substantial and reliable improvements in detecting unknown intrusions compared to the baseline methods. This strengthens the validity of our experimental results and supports the efficacy of our proposed approach.

5.7. Interpretation

Detecting intrusions in networks characterized by heterogeneous sources demands a multifaceted approach due to the varied nature of data streams.

Our study addresses this challenge by proposing a comprehensive solution encapsulated in a big universal security database. This database acts as a centralized repository, amalgamating data from disparate sources, including IoT devices, web applications, and web services, among others.

To effectively process this heterogeneous data, we establish three fundamental conditions that underpin our intrusion detection framework:

Heterogeneous Traffic: Given the diverse landscape of network communication, our approach necessitates the integration of multiple data sources to provide a holistic understanding of potential threats. This ensures that our model is robust and adaptable to the dynamic nature of network environments.

Table 8
Paired *t*-test results.

Metric	p-value	Statistical Significance
Accuracy	0.002	Yes
Precision	0.005	Yes
Recall	0.003	Yes
F1-score	0.001	Yes

Table 9
Wilcoxon signed-rank test results.

Metric	p-value	Statistical Significance
Accuracy	0.004	Yes
Precision	0.007	Yes
Recall	0.003	Yes
F1-score	0.002	Yes

Data Structure: Networks inherently exhibit varying data structures, making it imperative for our intrusion detection system to be capable of accommodating these differences. By embracing the diversity in data structures, our model can extract meaningful insights from disparate sources, enhancing its overall efficacy.

Detection System: Our intrusion detection system is designed to learn from and classify heterogeneous sources within a unified model. This unified approach enables the seamless integration of data from diverse sources, streamlining the detection process and improving overall accuracy.

While existing research has explored intrusion detection using diverse data sources, the focus has often been on analyzing individual datasets in isolation. In contrast, our study delves into the complexities of learning from data with disparate structures, a relatively unexplored domain in the literature.

A recent study by Xu, Shen and Du (2020) introduced a novel approach for network intrusion data reconstruction using a universal matrix. However, their evaluation primarily focused on datasets with similar structures, limiting the generalizability of their findings. In contrast, our proposed method demonstrates superior performance in handling data with diverse structures, as evidenced by our comprehensive evaluation.

In Table 10, we provide a comparative analysis between our approach and the method proposed in Xu, Shen and Du (2020), elucidating the strengths of our methodology in addressing the challenges posed by heterogeneous data sources. Our findings highlight the robustness and versatility of our approach, underscoring its potential to significantly enhance intrusion detection capabilities in complex network environments.

Table 10 presents a comparative analysis between the proposed method and Few-shot and Meta-learning approach (Xu, Shen & Du, 2020) regarding their performance in handling heterogeneous traffic and vector structures in intrusion detection. The Few-shot and Meta-learning approach achieved accuracies of 93.30% and 94.13% when learning jointly and separately on the CICIDS ISCX dataset, respectively. Similarly, for the CICIDS dataset, it attained an accuracy of 97.56%. In contrast, the proposed method achieved significantly higher accuracy of 99.80% when learning jointly on NSL KDD and CICIDS datasets, despite their different vector structures. This indicates the superiority of the proposed method in effectively handling heterogeneous traffic and varied vector structures, leading to enhanced intrusion detection accuracy.

In resume, our study contributes to advancing the field of intrusion detection by offering a unified framework capable of effectively

Table 10
Comparison of accuracy results with approaches using heterogeneous sources of net- work intrusion detection.

Methods	Heterogeneous traffic	Vector structure	Learning	%Accuracy
Few shot and meta learning (Xu, Shen & Du, 2020)	CICIDS ISCX	Same	Conjointly	93.30
Few shot and meta learning (Xu, Shen & Du, 2020)	CICIDS ISCX	Same	Separately	94.13 97.56
Proposed method	NSL KDD CICIDS	Different	Conjointly	**99.80**

processing heterogeneous data sources. By embracing the complexities of network traffic and data diversity, our approach lays the foundation for more resilient and adaptive cybersecurity measures in an increasingly interconnected world.

6. Discussion

While our proposed approach for heterogeneous intrusion detection demonstrates significant improvements in performance and scalability, it is important to acknowledge its potential limitations and scenarios where it may not perform well.

6.1. Potential weaknesses

1. Dependence on Feature Quality:
   - The effectiveness of our method relies heavily on the quality of the features extracted from the data. If the feature extraction process fails to capture critical characteristics of the intrusion patterns, the performance of the detection model may degrade.
2. Computational Complexity:
   - Although we have optimized the computational efficiency of our approach, the process of constructing the universal features vector and training ensemble models can still be resource-intensive for very large datasets. This may limit the scalability of the method in extremely large-scale environments.
3. Handling Concept Drift:
   - Our method assumes that the distribution of the data remains relatively stable over time. In real-world scenarios, the nature of intrusions can evolve, leading to concept drift. Our current approach may not adapt quickly to such changes, potentially affecting its long-term performance.
4. Imbalanced Data:
   - Intrusion detection datasets often suffer from class imbalance, where the number of normal instances significantly exceeds the number of intrusion instances. While our method includes mechanisms to handle imbalance, extreme cases of imbalance may still pose challenges and impact detection accuracy.
5. False Positives:
   - In an effort to maximize detection rates, our approach may generate false positives, flagging benign activities as intrusions. This can lead to increased workload for security analysts and may reduce the overall trust in the system.

6.2. Scenarios where the method may not perform well

1. Highly Dynamic Environments:
   - In environments where intrusion patterns change rapidly and significantly, our method may struggle to maintain high detection accuracy without frequent model updates and retraining.
2. Sparse Data:
   - In scenarios where the available data is sparse or incomplete, the feature extraction process may not yield meaningful representations, thereby affecting the overall performance of the detection model.
3. Highly Encrypted Traffic:
   - Our approach relies on analyzing features extracted from network traffic. If the traffic is highly encrypted, it may be challenging to extract meaningful features, reducing the efficacy of our method.
4. Adversarial Attacks:
   - Our method may be vulnerable to adversarial attacks where attackers deliberately manipulate data to evade detection. Robustness against such attacks is an area that requires further research and improvement.

6.3. Future work

To address these limitations, future research could focus on:

1. Enhancing Feature Extraction:
   - Developing more sophisticated feature extraction techniques that can capture complex and evolving intrusion patterns.
2. Adaptive Models:
   - Implementing adaptive models that can dynamically update themselves in response to changes in data distribution and intrusion patterns.
3. Reducing False Positives:
   - Exploring advanced techniques to reduce false positives without compromising detection accuracy.
4. Handling Imbalanced Data:
   - Investigating novel methods to better handle extreme class imbalance in intrusion detection datasets.
5. Robustness Against Adversarial Attacks:
   - Enhancing the robustness of the detection models to withstand adversarial attacks and ensuring reliable performance in adversarial settings.

By critically analyzing these potential weaknesses and scenarios, we aim to provide a comprehensive understanding of the limitations of our proposed approach and identify areas for future improvement.

6.4. Novelty and contributions

While our proposed approach leverages established techniques such as TF-IDF, Cosine similarity, PCA, and classical machine learning algorithms, the novelty lies in the innovative integration and application of these techniques within a unified framework. This integration addresses specific challenges in heterogeneous intrusion detection and results in notable advancements over existing methodologies.

6.4.1. Unique aspects of our approach

1. Universal Features Vector:
   - The construction of a universal features vector that captures essential characteristics from diverse datasets is a key innovation. By integrating TF-IDF, Cosine similarity, and PCA, we create a robust feature representation that enhances the detection of unknown intrusions across heterogeneous data sources.
2. Ensemble Learning for Heterogeneous Data:
   - Our approach combines Random Forest and Adaboost in a novel ensemble learning framework tailored for heterogeneous intrusion detection. This ensemble effectively balances the strengths of each algorithm, improving overall detection accuracy and robustness.
3. Scalability and Efficiency:
   - We optimize the computational efficiency of our approach, making it practical for real-time intrusion detection in large-scale environments. The efficient construction of the universal features vector and the scalable nature of ensemble learning contribute to the framework's applicability in real-world scenarios.
4. Enhanced Detection of Unknown Intrusions:
   - Our method focuses on detecting unknown intrusions by leveraging the universal features vector and ensemble learning. This emphasis on unknown intrusion detection addresses a critical gap in existing methodologies that often rely on predefined signatures or patterns.

6.4.2. Contributions

1. Innovative Integration of Established Techniques:

- We demonstrate that the innovative integration of well-established techniques within a unified framework can lead to significant performance improvements in heterogeneous intrusion detection.

2. Comprehensive Evaluation:
   - Our extensive experimental evaluation, including statistical significance testing and comparison with state-of-the-art approaches, validates the effectiveness and superiority of our method.
3. Practical Implementation:
   - We provide detailed pseudocode and implementation guidelines, facilitating the reproducibility and practical adoption of our approach in various intrusion detection systems.
4. Addressing Real-World Challenges:
   - Our approach effectively addresses real-world challenges such as handling high-dimensional data, balancing bias-variance trade-offs, and improving detection accuracy and robustness.

By highlighting these unique aspects and contributions, we aim to clarify the novelty of our approach and demonstrate its significant advancements over existing methodologies.

7. Conclusion and future work

In this study, we validated the hypothesis that a comprehensive ensemble learning approach, integrating heterogeneous datasets, can significantly enhance the detection of unknown intrusions. By applying various machine learning algorithms across the NSL-KDD, UNSW, and CICIDS datasets, we demonstrated the high detection accuracy and generalizability of our method, supporting our claim that ensemble learning can effectively address the challenges posed by diverse data environments in intrusion detection.

While our results are promising, some limitations remain. The datasets used, though extensive, may not encompass all types of network intrusions, highlighting the need for more diverse and recent datasets to maintain model robustness against evolving threats. Our feature extraction process relies on techniques like TF-IDF and Cosine similarity, which, while effective, could be complemented by advanced feature engineering to further improve detection accuracy and reduce potential biases. Additionally, the combination of parallel and sequential ensemble learning approaches, though beneficial in accuracy, increases computational demands, potentially impacting the method's applicability in resource-constrained, real-time systems.

Future research could address these limitations by incorporating more varied datasets, exploring automated feature engineering methods (e.g., deep learning-based extraction), and optimizing the computational complexity of the model. Integrating our approach into real-time intrusion detection systems presents another valuable direction for further study, supporting broader deployment.

By recognizing these limitations and suggesting future improvements, this study contributes to a more robust and scalable framework for intrusion detection, reinforcing the effectiveness and relevance of ensemble learning for securing complex network environments.

Credit author statement

The paper titled "Detecting Unknown Intrusions from Large Heterogeneous Data through Ensemble Learning" was authored by Farah Jemili, Khaled Jouini, and Ouajdi Korbaa. Farah Jemili contributed as the first author, leading the conception, design, and execution of the research presented in this paper. Khaled Jouini assisted significantly in providing substantial input during the revision process. Ouajdi Korbaa also made significant contributions to the research design and execution. Farah Jemili took the lead in drafting the manuscript, with critical revisions and input from Khaled Jouini and Ouajdi Korbaa. All authors have approved the final version of the manuscript and agree to be accountable for all aspects of the work.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

Arasteh, B., Aghaei, B., Farzad, B., Arasteh, K., Kiani, F., & Torkamanian-Afshar, M. (2024). Detecting SQL injection attacks by Binary Gray Wolf optimizer and machine learning algorithms. *Neural Computing and Applications, 36*, 6771–6792. https://doi.org/10.1007/s00521-024-08001-5

Arasteh, B., Bouyer, A., Sefati, S. S., & Craciunescu, R. (2024). Effective SQL injection detection: A fusion of binary Olympiad optimizer and classification algorithm. *Mathematics, 12*(18), 2917. https://doi.org/10.3390/math12182917

ARGUS IDS: https://openargus.org/argus-ml/2-uncategorised/30-unsw-nb15, last accessed 2022/02/20.

CIC Flow Meter: https://github.com/ahlashkari/CICFlowMeter, last accessed 2022/02/20.

CICIDS 2017: https://www.unb.ca/cic/datasets/ids-2017.html, last accessed 2022/02/01.

Danesh, H., Karimi, M. B., & Arasteh, B. (2024). CMShark: A NetFlow and machine-learning based Crypto-Jacking intrusion-detection method. *Intelligent Decision Technologies, 18*(3), 2255–2273. https://doi.org/10.3233/IDT-240319

Elayni, M., Jemili, F., Korbaa, O., & Soulaimen, B. (2019). Big Data processing for intrusion detection system context: A review. In *The International Conference on Intelligent Systems Design and Applications (ISDA) At: Pretoria, South Africa*.

Intrata, Evgeniya, Grif, Mikhail, & Dostovalov, Dmitry (2021). Application of traditional machine learning models to detect abnormal traffic in the internet of things networks. In *International Conference on Computational Collective Intelligence ICCC*. Cham: Springer.

NSL-KDD: https://github.com/defcom17/NSL_KDD, last accessed 2022/02/01.

Othman, Suad Mohammed (2018). Intrusion detection model using machine learning algorithm on Big Data environment. *Journal of big data, 5*, 1–12.

Patel, A., Taghavi, M., Bakhtiyari, K., & Júnior, J. C. (2013). An intrusion detection and prevention system in cloud computing: A systematic review. *Journal of Network and Computer Applications, 36*(1), 25–41.

Peng, kai, Leung, Victor C. M., & Huang, Qingjia (2018). Clustering approach based on mini batch kmeans for intrusion system over big data. *IEEE Access, 6*, 11897–11906.

Xu, Congyuan, Shen, Jizhong, & Du, Xin (2020). A method of few-shot network intrusion detection based on meta-learning framework. *IEEE Transactions on Information Forensics and Security, 1*, 3540–3552.

Zhou, Donghao, et al. (2018). A survey on network data collection. *Journal of Network and Computer Applications, 11*, 9–23.

Zuech, Richard, Khoshgoftaar, Taghi M., & Wald, Randall (2015). Intrusion detection and big heterogeneous data: A survey. *Journal of Big Data, 2*, 1–41.

Dr. Farah JEMILI had the Engineer degree in Computer Science in 2002, the Master degree in 2004, and the Ph.D degree in 2010 from the National School of Computer Science (ENSI, University of Manouba, Tunisia). Since 2007, she is an Assistant Professor at the Higher Institute of Computer Science and Telecom of Hammam Sousse (ISITCOM, University of Sousse, Tunisia). She started research since 2002 at RIADI Laboratory (ENSI, University of Manouba, Tunisia). Since 2010, she is a Senior Researcher at MARS Laboratory (ISITCOM, University of Sousse, Tunisia). Her research interests include Artificial Intelligence, Cyber Security, Big Data Analysis and Distributed Systems. She served as a Reviewer for many international conferences and journals. She has published around 45 Research papers in international journals and conferences and has presented many invited and contributed Talks at international conferences. https://orcid.org/0000-0001-7511-1221

Dr. Khaled JOUINI Khaled Jouini is an accomplished Assistant Professor at ISITCOM, University of Sousse, Tunisia, with over 14 years of academic experience. He has also worked as an IT consultant for Sam's-Tech and held various roles at Université Paris Dauphine. Khaled holds a Doctorate in Computer Science from Université Paris Dauphine and is certified as a Big Data Specialist with IBM BigInsights. His expertise spans Apache Spark and database management, and he has taught courses on Oracle Database Administration and Advanced Databases at ISITCom. Additionally, Khaled has served as a reviewer for numerous international conferences and journals, and he has published many research papers in prominent international journals and conferences. https://orcid.org/0000-0001-5049-4238

Pr. Ouajdi KORBAA obtained in 1995 the Engineering Diploma from the Ecole Centrale de Lille (France), and in the same year, the Master degree in Production Engineering and Computer Sciences from the University of Lille I. He is Ph.D. in Production Management, Automatic Control and Computer Sciences of the University of Sciences and Technologies of Lille (France) since 1998. He also obtained, from the same university, the Habilitation to Supervise Researches degree in Computer Sciences in 2003. He is full Professor in the University of Sousse. He published around 150 research papers on scheduling, performance evaluation, discrete optimization, design, and monitoring. https://orcid.org/0000-0003-4462-1805

.

Q2.1 On the Use of Text Augmentation For Stance and Fake News Detection

Ilhem Salah, Khaled Jouini & Ouajdi Korbaa
Journal of Information and Telecommunication. 2023.
Vol. 7, No. 3, pp 359-375, ISSN : 24751839, Taylor and Francis Ltd.
JCR IF : 2.7

DOI : https://doi.org/10.1080/24751839.2023.2198820
SJR best quartile : Q2, SJR : 0.67

Journal of Information and Telecommunication

COUNTRY	SUBJECT AREA AND CATEGORY	PUBLISHER	H-INDEX
United Kingdom Universities and research institutions in United Kingdom Media Ranking in United Kingdom	Computer Science Computer Networks and Communications Computer Science Applications Computer Science (miscellaneous) Engineering Electrical and Electronic Engineering	Taylor and Francis Ltd.	16

PUBLICATION TYPE	ISSN	COVERAGE	INFORMATION
Journals	24751839, 24751847	2017-2023	Homepage How to publish in this journal nguyenngocthanh@tdtu.edu.vn

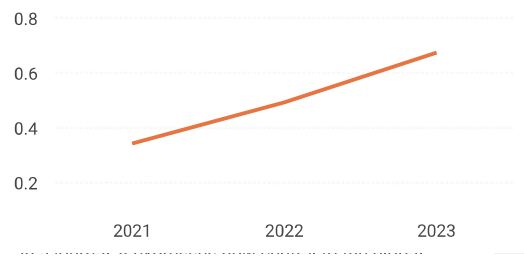
SCOPE

The Journal of Information and Telecommunication (JIT) is an open access peer-reviewed journal that publishes original research on all aspects of telecommunications, information technology, media technologies and media communications. JIT represents the process of the convergence of the fields of telecommuni-cations, digital broadcasting and information technology. The journal focuses on the field of infocommunication as a natural expan-sion of telecommunication, with information processing and content handling functions which include all types of electronic communications. In particular, the journalaims to address infocommunication issues which are unique to developing nations. For example, research topics on smart tele-communication systems for developing countries, applications for proces-sing Asian languages in telecommunication systems etc., are encouraged. Topics of interest to the journal include, but are not limited to: • Information theory/coding, network security, standards, applications, and information processing in telecommunication systems • Internet/web-based systems/products, internet of things • Network interconnection, wire, wireless, adhoc, mobile networks • Cost benefit analysis and economic impact of telecommunication sys-tems • Standardization and regulatory issues • Security, privacy and encryption in telecommunication systems • Cellular, mobile and satellite-based systems • Intelligent information systems, intelligent information retrieval, digital libraries, and network information retrieval • Visual interfaces, visual query languages, and visual expressiveness of IIS • Machine learning, knowledge discovery, and data mining

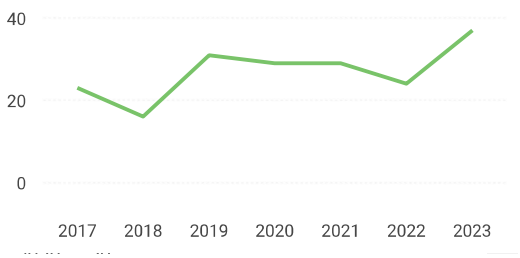
Join the conversation about this journal

Quartiles

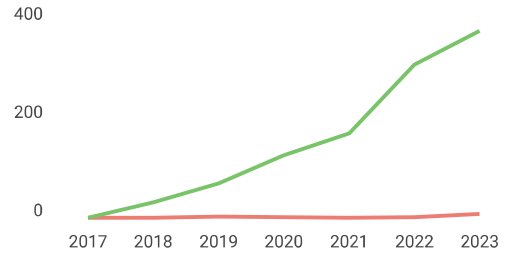
SJR



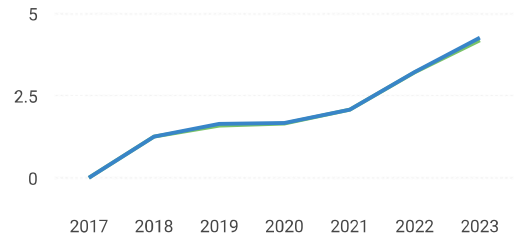
Total Documents



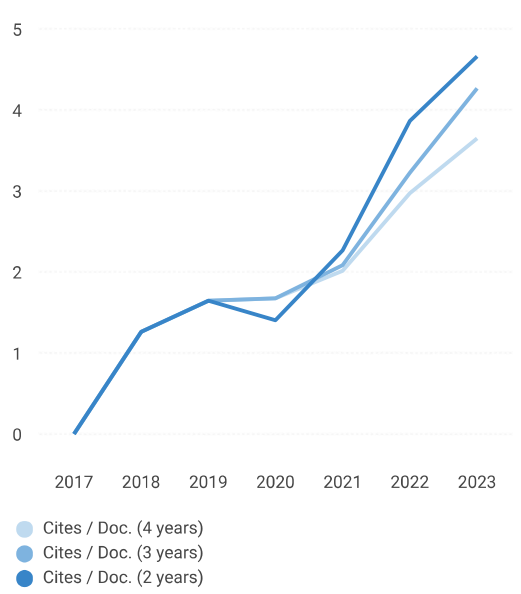
Total Cites Self-Cites



External Cites per Doc Cites per Doc

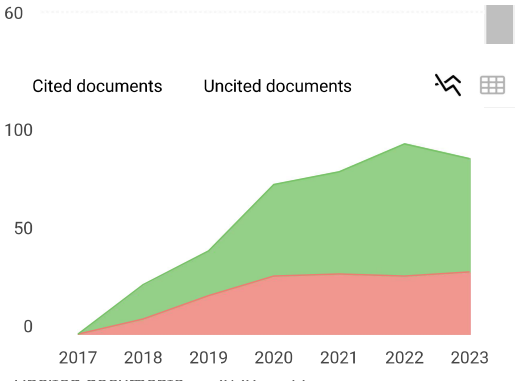


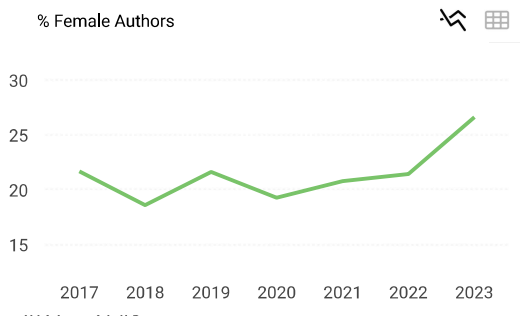
Citations per document

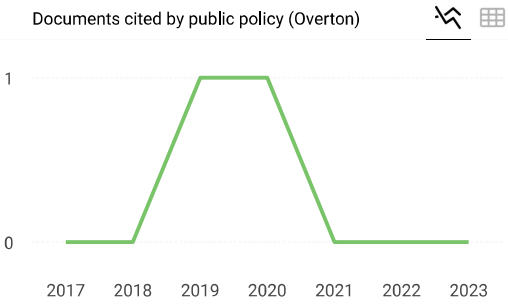


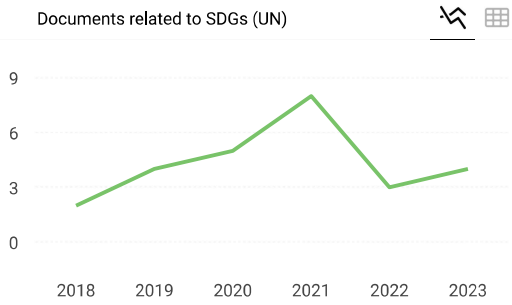
% International Collaboration

Citable documents Non-citable documents









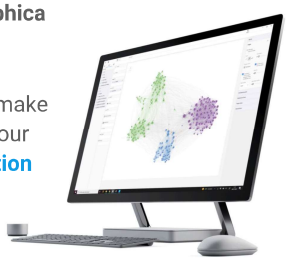
← Show this widget in your own website

Just copy the code below and paste within your html code:

```
<a href="https://www.scimag
```

SCImago Graphica

Explore, visually communicate and make sense of data with our **new data visualization tool**.



Metrics based on Scopus® data as of March 2024

JOURNAL OF INFORMATION AND TELECOMMUNICATION
2023, VOL. 7, NO. 3, 359–375
https://doi.org/10.1080/24751839.2023.2198820

On the use of text augmentation for stance and fake news detection

Ilhem Salah, Khaled Jouini and Ouajdi Korbaa

MARS Research Lab LR17ES05, ISITCom, University of Sousse, H. Sousse, Tunisia

ABSTRACT

Data Augmentation (DA) aims at synthesizing new training instances by applying transformations to available ones. DA has several well-known benefits such as: (i) increasing generalization ability; (ii) preventing data scarcity; and (iii) helping resolve class imbalance issues. In this work, we investigate the use of DA for stance and fake news detection. In the first part of our work, we explore the effect of various DA techniques on the performance of common classification algorithms. Our study reveals that the motto '*the more, the better*' is the wrong approach regarding text augmentation and that there is no *one-size-fits-all* text augmentation technique. The second part of our work leverages the results of our study to propose a novel augmentation-based, ensemble learning approach. The proposed approach leverages text augmentation to enhance base learners' diversity and accuracy, ergo the predictive performance of the ensemble. The third part of our work experimentally investigates the use of DA to cope with the class imbalance problem. Class imbalance is very common in stance and fake news detection and often results in biased models. In this work we show how and to what extent text augmentation can help resolving moderate and severe imbalance.

ARTICLE HISTORY
Received 30 December 2022
Accepted 26 March 2023

KEYWORDS
Stance and fake news detection; text augmentation; ensemble learning; class imbalance

1. Introduction

In the era of the Internet and social media, where a myriad of information of various types is instantly available and where any point of view can find an audience, access to information is no longer an issue, and the key challenges are veracity, credibility, and authenticity. The reason for this is that any user can readily gather, consume, and break news, without verification, fact-checking, or third-party filtering.

By directly influencing public opinions, major political events, and societal debates, fake news has become the scourge of the digital era, and combating it has become a dire need. The identification of fake news is however very challenging, not only from a machine learning and Natural Language Processing (NLP) perspective, but also

CONTACT Ilhem Salah ilhemsalah53@gmail.com MARS Research Lab LR17ES05, ISITCom, University of Sousse, H. Sousse 4011, Tunisia

sometimes for the most experienced journalists (Pomerleau & Rao, 2017). That is why the scientific community approaches the task from a variety of angles and often breaks down the process into independent sub-tasks. A first practical step towards automatic fact-checking and fake news detection is to estimate the opinion or the point of view (i.e. *stance*) of different news sources regarding the same topic or claim (Pomerleau & Rao, 2017). This (sub-) task, addressed in recent research as *stance detection*, was popularized by the Fake News Challenge – Stage 1 (or FNC-1) (Pomerleau & Rao, 2017), which compares article bodies to article headlines and determines if a body agrees, disagrees, discusses or is unrelated to the claim of a headline. As aptly stated by Momchil et al. (2022), automated stance detection can help in identifying fake news in two key ways. First, it enables human fact-checkers to quickly and efficiently, identify controversial claims, gather relevant opinions about a claim, and evaluate the arguments for and against it (i.e. *evidence retrieval*). Second, it can be integrated as a component of an automated fact-checking pipeline, which would give a preliminary label to a claim, based on the stances taken by various sources, weighted by their credibility (Guo et al., 2021).

In a previous work (Salah et al., 2022), we proposed a novel *Augmentation-based Ensemble learning* approach for stance and fake news detection. Data augmentation aims at synthesizing new training instances that have the same ground-truth labels as the instances that they originate from (Xie et al., 2019). Data augmentation has several well-known benefits: (i) preventing overfitting by *improving the diversity* of training data; (ii) *preventing data scarcity* by providing a relatively easy and inexpensive way to collect and label data; (iii) increasing the *generalization ability* of the obtained models; and (iv) helping resolve *class imbalance* issues. Data augmentation is extensively used in Computer Vision (CV) where it is considered as one of the anchors of good predictive performance. Despite promising advances, data augmentation remains however less explored in NLP where it is still considered as the 'cherry on the cake' which provides a steady but limited performance boost (Shorten et al., 2021).

Ensemble learning combines the knowledge acquired by base learners to make a consensus decision which is supposed to be superior to the one attained by each base learner alone (Suting & Ning, 2020). Research on ensemble learning proves that the greater are the skills and the diversity of base learners, the better are the accuracy and the generalization ability of the ensemble (Suting & Ning, 2020). In our work we leverage text augmentation to enhance both, the diversity and the skills of base learners, ergo the predictive performance of the ensemble.

Class imbalance refers to situations where the distribution of examples across the classes is not equal, i.e. the number of examples available for one or more classes (minority classes) is far less than other classes (majority classes). Class imbalance appears in many domains, including fraud detection, disease screening and fake news detection. When a dataset is imbalanced, most classifiers have *a strong bias toward majority classes* (Fernndez et al., 2018). This paper extends our previous work (Salah et al., 2022) by presenting more comprehensive experimental results, additional related work, and deeper insights into our augmentation-based ensemble approach. Furthermore, this paper evaluates the effectiveness of text augmentation in addressing severe and moderate class imbalance, a common issue in stance and fake news detection, not explored in (Salah et al., 2022).

The main contributions of our work are therefore: (i) an extensive experimental study on the effect of different text data augmentation techniques on the performance of common classification algorithms; (ii) a novel augmentation-based ensemble learning approach; and (iii) an experimental study on the use of text augmentation to mitigate the effects of class imbalance.

The remainder of this paper is organized as follows. Section 2 outlines the main steps we followed to vectorize text and reduce dimensionality. Section 3 exposes the key motifs of data augmentation and the text augmentation techniques adopted in our work. Section 4 details the architecture of our novel augmentation-based ensemble learning. Section 5 briefly reviews existing work on stance and fake news detection. Section 6 presents an experimental study on two real-world fake news datasets and discusses the main results and findings. Finally, Section 7 concludes the paper.

2. Text as vectors

2.1. Pre-processing and feature extraction

Machine Learning (ML) algorithms operate on numerical features, expecting input in the form of a matrix where rows represent instances and columns features. Raw texts have therefore to be transformed into feature vectors before feeding into ML algorithms (Jouini et al., 2021). In our work, we first eliminated stop words and reduced words to their roots (i.e. base words) by stemming them using Snowball Stemmer from the NLTK library (NLTK.org., n.d.). We next vectorized the corpus with a TF-IDF (*Term Frequency – Inverse Document Frequency*) weighting scheme and generated a term-document matrix.

TF–IDF is computed on a per-term basis, such that the relevance of a term to a text is measured by the scaled frequency of the appearance of the term in the text, normalized by the inverse of the scaled frequency of the term in the entire corpus. Despite its simplicity and its wide-spread use, the TF-IDF scheme has two severe limitations: (i) TF-IDF does not capture the co-occurrence of terms in the corpus and makes no use of semantic similarities between words. Accordingly, TF-IDF fails to capture some basic linguistic notions such as synonymy and homonymy; and (ii) The term-document matrix is high dimensional and is often noisy, redundant, and excessively sparse. The matrix is thus subject to the curse of dimensionality: as the number of features is large, poor generalization is to be expected.

2.2. Dimensionality reduction

Latent Semantic Analysis (LSA) (Deerwester et al., 1990) is an unsupervised statistical topic modelling technique, overcoming some of the limitations of TF-IDF. As other topic modelling techniques, such as LDA (Latent Dirichlet Allocation (Blei et al., 2003)), LSA is based on the assumptions that: (i) each text consists of a mixture of *topics*; and (ii) each topic consists of a set of (weighted) terms that regularly co-occur together. Put differently, the basic assumption behind LSA is that words that are close in meaning, appear in similar contexts and form a 'hidden topic'. The idea behind LSA is then to represent words that form a topic not as separate dimensions, but by a single dimension. LSA

represents thus texts by 'semantic' or 'topic' vectors, based on the words that these texts contain and the set of weighted words that form each of the topics.

To uncover the latent topics that shapes the meaning of texts, LSA performs a Singular Value Decomposition (SVD) on the document-term matrix (i.e. decomposes it into a separate text-topic matrix and a topic-term matrix). Formally, SVD decomposes the term-document matrix $A_{t\times n}$, with t the number terms and d the number of documents, into the product of three different matrices: orthogonal column matrix, orthogonal row matrix and one singular matrix.

$$A_{t\times n} = U_{t\times n} S_{n\times n} D^{T}_{n\times d} \tag{1}$$

where $n = \min(t, d)$ is the rank of A. By restricting the matrices T, S and D to their first $k<n$ rows, we obtain the matrices $T_{t\times k}$, $S_{k\times k}$ and $D_{d\times k}$, and hence obtain k-dimensional text vectors. From a practical perspective, the key ask is to determine k, which would be reasonable for the problem (i.e. without major loss). In our work we used the transformer TruncatedSVD from sklearn (Pedregosa et al., 2011). As in Li et al. (2019) we set the value of k to 100D. The experimental study conducted in Li et al. (2019) showed that using LSA (with k set to 100D) instead of TF-IDF allows a substantial performance improvement for the tasks of stance and fake news detection.

3. Text data augmentation

The success of data augmentation in Computer Vision has been fuelled by the ease of designing semantically invariant transformations (i.e. label-preserving transformations), such as rotation, flipping, etc…While recent years witnessed significant advancements in the design of transformation techniques, text augmentation remains less explored and adopted in NLP than in CV. This is mainly due to the intrinsic properties of textual data (e.g. polysemy), which make defining label-preserving transformations much harder (Shorten et al., 2021). In the sequel we mainly focus on off-the-shelf text augmentation techniques and less on techniques that are still in the research phase, waiting for large-scale testing and adoption. For a more exhaustive survey on text augmentation techniques, we refer the reader to Karnyoto et al. (2022), Li et al. (2021), and Tesfagergish et al. (2021).

3.1. Masked language models

The main idea behind *Masked Language Models* (MLMs), such as BERT (Devlin et al., 2018), is to mask words in sentences and let the model predict the masked words. BERT, which is a pretrained multi-layer bidirectional transformer encoder, has the ability to predict masked words based on the bidirectional context (i.e. based on its left and right surrounding words). In contrast with other context-free models such as GLOVE and Word2Vec, BERT alleviates the problem of ambiguity since it considers the whole context of a word.

BERT is considered as a breakthrough in the use of ML for NLP and is widely used in a variety of tasks such as classification, Question/Answering, and Named Entity Recognition (Shi et al., 2020). Inspired by the recent work of Li et al. (2021) and Shi et al. (2020), we use BERT as an augmentation technique. The idea is to generate new sentences by randomly masking words and replacing them by those predicted by BERT.

3.2. Back-translation (*a.k.a.* round-trip translation)

Back-Translation is the process of translating a text into another language, then translating the new text back into the original language. Back-translation is one of the most popular means of paraphrasing and text augmentation (Marivate & Sefara, 2019). Google Cloud Translation API, used in our work to translate sentences to French and back, is considered as the most common tool for back-translation (Li et al., 2021).

3.3. Synonym (*a.k.a.* thesaurus-based augmentation)

The *Synonym* technique, also called lexical substitution with dictionary, was until recently the most widely (and for a long time the only) augmentation technique used for textual data classification. As suggested by its name, the Synonym technique replaces randomly selected words with their respective synonyms. The types of words that are candidates for lexical substitution are: adverbs, adjectives, nouns and verbs.

The synonyms are typically taken from a lexical database (i.e. dictionary of synonyms). WordNet (Shoemaker, 2019), used in our work for synonym replacement, is considered as the most popular open-source lexical database for the English language (Li et al., 2021).

3.4. TF-IDF based insertion and substitution

The intuition behind these two noising-based techniques is that uninformative words (i.e. words having low TF-IDF scores) should have no or little impact on classification. Therefore, the insertion of words having low TF-IDF scores (at random positions) should preserve the label associated with a text, even if the semantics are not preserved. An alternate strategy is to replace randomly selected words with words having the same low TF-IDF scores (*TF-IDF based substitution*).

Section 6 presents an extensive study on the effect of the aforementioned augmentation techniques on the prerediction performance of ten common classification algorithms, namely, Decision Tree (DT), Support Vector Machine (SVM), Adaptive Boosting (AdaBoost), Random Forest (RF), eXtreme Gradient Boosting (XGBoost), Bagged Random Forests (Bagged RF), Light Gradient Boosting Machine (LightGBM), Gradient Boosting (GradBoost), Logistic Regression (LR), and Naive Bayes (NB).

4. Augmentation-based ensemble learning

4.1. Diversity and skilfulness in ensemble learning

Ensemble Learning finds its origins in the 'Wisdom of Crowds' theory (Surowiecki, 2005). The 'Wisdom of Crowds' theory states that the collective opinion of a group of individuals can be better than the opinion of a single expert, provided that the aggregated opinions are diverse (i.e. diversity of opinion) and that each individual in the group has a minimum level of competence (e.g. better than a random guess). Similarly, Ensemble Learning combines the knowledge acquired by a group of base learners to make a consensus decision which is supposed to be superior to the one reached by each of them separately (Suting & Ning, 2020). Research on Ensemble Learning proves that the greater are the skills and the diversity of base models, the better is the generalization ability of the ensemble model (Suting & Ning, 2020). Alternatively stated, to generate a good ensemble model, it is

necessary to build base models that are, not only skilful, but also skilful in a different way from one another.

Bagging and stacking are among the main classes of parallel ensemble techniques. *Bagging* (i.e. Bootstrap aggregating) involves training multiple instances of the same classification algorithm, then combining the predictions of the obtained models through hard or soft voting. To promote diversity, base learners are trained on different subsets of the original training set. Each subset is typically obtained by drawing random samples with replacement from the original training set (i.e. bootstrap samples). *Stacking* (*a.k.a.* stacked generalization) involves training a learning algorithm (i.e. meta-classifier) to combine the predictions of several heterogeneous learning algorithms, trained on the same training data. The most common approach to train the meta-model is via k-fold cross-validation. With the k-fold cross-validation, the whole training dataset is randomly split (without replacement) into independent equal-sized k-folds. $k-1$ folds are then used to train each of the base models and the kth fold (holdout fold) is used to collect the predictions of base models on unseen data. The predictions made by base models on the holdout fold, along with the expected class labels, provide the input and the output pairs used to train the meta-model. This procedure is repeated k times. Each time a different fold acts as the holdout fold while the remaining folds are combined and used for training the base models.

4.2. Novel augmentation-based approach

As mentioned earlier, in conventional stacking base learners are trained on the same dataset and diversity is achieved by using heterogeneous classification algorithms. As depicted in Figure 1, the classical approach for combining augmentation and stacking, is to: (i) apply one or several augmentation techniques to the original dataset, (ii) fuse the original dataset and data obtained through augmentation; and (iii) train base learners on the fused dataset. In our work we adopt a different approach and train heterogeneous algorithms on different data to further promote diversity. More specifically, through an extensive experimental study (Section 6), we first identify the most accurate *(augmentation technique, classification algorithm)* pairs. Our meta-model is then trained on the predictions

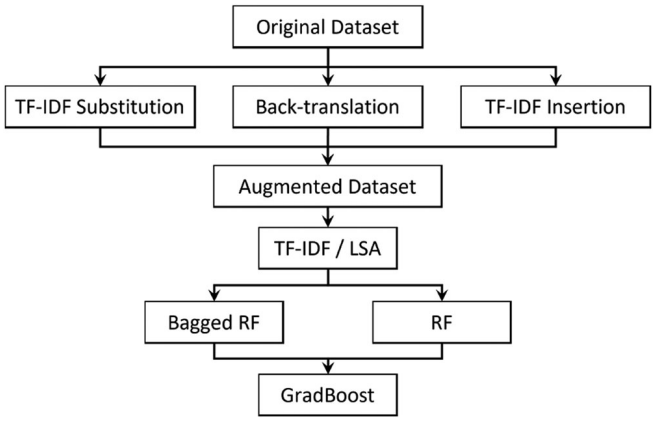


Figure 1. Conventional approach for combining augmentation and stacking.

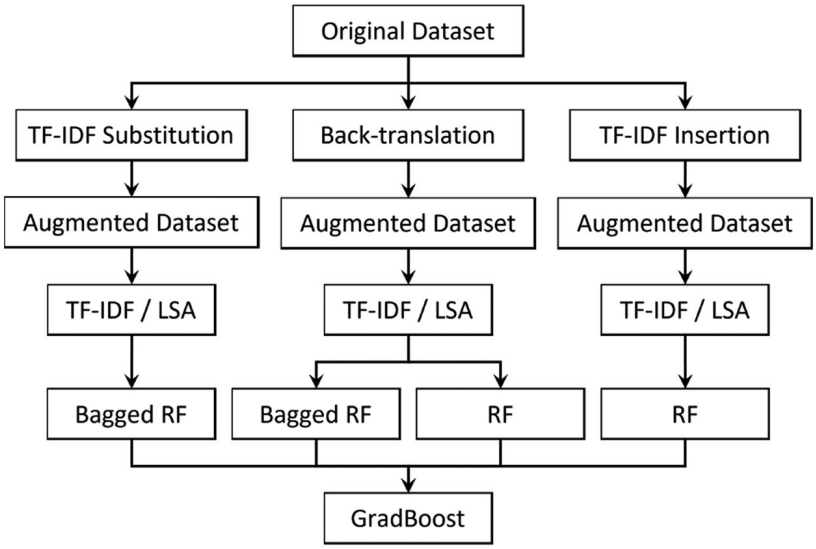


Figure 2. Novel augmentation-based ensemble learning approach.

made by the most accurate pairs, using a stratified k-fold cross-validation. Figure 2 depicts the overall architecture of the proposed augmentation-based ensemble learning.

Our augmentation-based ensemble learning approach, can be seen as a mixture between stacking and bagging. In contrast with Bagging and like Stacking, we use an ensemble of heterogeneous learning algorithms. In contrast with stacking and like Bagging, base learners are trained on different datasets. However, unlike Bagging the considered datasets are not obtained through bootstrap sampling. Instead, they are obtained by combining the original training data with the data obtained by applying one of the aforementioned text augmentation techniques. Finally, like in conventional Stacking, the meta-model is trained using a stratified K-fold cross-validation.

5. Related work

Salient stance and fake news detection approaches adopt a wide range of different features (e.g. context-based, content-based), classifiers, and learning tactics (e.g. stacking, bagging, etc.). Due to the lack of space, we mainly focus hereafter on ensemble approaches and on approaches that rely on content-based features. We suggest readers to refer to surveys and retrospectives on recent challenges (Hanselowski et al., 2018; Khan et al., 2021) for a more comprehensive overview of the current state of research. In the sequel, we distinguish between approaches dedicated to stance classification (*multinomial classification*) and those intended to fake news classification (*binary classification*).

5.1. Stance classification

The authors of the fake news challenge (FNC-1) (Slovikovskaya, 2019), released a simple baseline model for the stance detection task. The proposed model achieves an

F1-score of 79.53% and uses a gradient boosting (GradBoost) classifier on global co-occurrence, polarity and refutation features. The three best performing systems in the FNC-1 competition were 'SOLAT in the SWEN' (Pan, 2018), 'Team Athene' (Hanselowski et al., 2018) and 'UCL Machine Reading' (UCLMR) (Riedel et al., 2017). 'SOLAT in the SWEN' won the competition using an ensemble approach based on a 50/50 weighted average between gradient-boosted decision trees and a Convolutional Neural Network (CNN). The proposed system is based on several features: Word2Vec pretrained embeddings, TF-IDF, Single Value Decomposition and WordCount. The convolutional network uses pre-trained Word2Vec embeddings passed through several convolutional layers followed by three fully-connected layers and a final softmax layer for classification. Hanselowski et al. (2018), the second place winner, used an ensemble composed of 5 Multi-Layer Perceptrons (MLPs), where labels are predicted through hard voting. The system of UCLMR (Riedel et al., 2017), placed third, used an MLP classifier with one hidden layer of 100 units and a softmax layer for classification.

Recently, other published work used FNC-1 in their experiments.In particular, several recent approaches (Dulhanty et al., 2019; Sepúlveda-Torres et al., 2021; Slovikovskaya, 2019) construct stance detection language models by performing transfer learning on pre-trained variants of BERT (mainly BERT, RoBERTa and XLNet).

Among these approaches, Sepúlveda-Torres et al. (2021) stands out for its integration of *text summarization*. Text summarization involves reducing a long text into a shorter version, while preserving its most important information. From an overarching perspective, text summarization and text augmentation can both be seen as techniques that alter the form of a text (making it more concise or more diverse) to better capture its core meaning.

The approach proposed by Sepúlveda-Torres et al. (2021) involves two stages: *Relatedness Stage* and *Stance Stage*. The Relatedness Stage is in charge of determining whether or not a headline and a news summary are related. This stage uses TextRank extractive algorithm for body text summarization and a fine-tuned RoBERTa pre-trained model that classifies a headline-summary pair as related or unrelated. Once the related pairs are identified, the Stance stage determines their type with respect to the remaining stances (*agree*, *disagree* or *discuss*). Similarly to the Relatedness Stage, the Stance stage uses a fine-tuned RoBERTa pre-trained model to yield predictions. While not presented as such by the authors, we believe that by discarding unrelated pairs (the majority class) at an early stage of the process, the approach of Sepúlveda-Torres et al. (2021) partially resolves the class imbalance problem. The work of Sepúlveda-Torres et al. (2021) on text summarization and ours on text augmentation, both demonstrate that modifying the form of texts while preserving their respective meanings can result in improved predictive performance. A downside of the work of Sepúlveda-Torres et al. (2021) is that it is not suitable for short text messages, which are prevalent on social media.

5.2. Fake news classification

Besides stance detection, several ensemble learning models have been proposed to tackle the binary (*True News*/*False News*) content-based classification task. Notably, Jiang et al. (2021) proposed a stacking-based ensemble that uses Random Forest (RF) as meta-learner and Support Vector Machine (SVM), Logistic Regression (LR), Decision

Tree (DT), k-nearest neighbours (KNN), Random Forest (RF), Convolution Neural Network (CNN), Long short-term memory (LSTM) and Gated Recurrent Network (GRU) as base learners. The approach of Jiang et al. (2021) uses three different text vectorization methods: Word2Vec embedding, TF-IDF and TF. The proposed approach was evaluated on ISOT fake news (Ahmed et al., 2017) and KDnuggets (McIntire, 2017) datasets. Similarly, Patil (2022) proposed a majority voting ensemble model involving nine base learners, namely: SVM, DT, LR, RF, X-Gradient Boosting (XGBoost), Extra Trees (ET), AdaBoost, Stochastic Gradient Descent (SGD) and Naive Bayes (NB). The proposed approach was evaluated on Kaggle Fake News dataset (Kaggle.com, n.d.). In the same vein as Patil (2022) and Mahabub (2020) uses a majority voting classifier with an ensemble composed of three base learners, namely, MLP, LR and XGBoost. Mahabub (2020) experimented their approach on the dataset LIAR proposed by Wang (2017).

The work of Li et al. (2019), which is the closest to the spirit of our work, uses LSA for dimensionality reduction and a stacking-based ensemble having five base learners: GradBoost, Random Forest (RF), XGBoost, Bagging and Light Gradient Boosting Machine (Lightgbm). Besides, Li et al. (2019) compared LDA and LSA and found that LSA yields better accuracy. The authors in Li et al. (2019) experimented their approach on FNC-1 and FNN datasets. Li et al. (2019) has not addressed the issue of class imbalance. An experimental comparison between (Li et al., 2019) and our work is given in Section 6.

It is worth noticing that in all the aforementioned studies, ensemble approaches yielded better results than those attained by their contributing base learners. On the other hand, despite the substantial potential improvement that text augmentation can carry out, to the best of our knowledge, there exists no previous work on stance and fake news detection that compares text augmentation techniques, uses text augmentation in conjunction with ensemble learning or mitigates the effects of class imbalance through text augmentation.

6. Experimental study

6.1. Tools & datasets

Our system was implemented using NLTK (NLTK.org., n.d.) for text preprocessing, nlpaug (Ma, 2019) for text augmentation, SciKit-Learn (version 0.24.2) (Pedregosa et al., 2011) for classification and Beautiful Soup for web scraping. A stratified 10-fold cross-validation was used for model fusion. The Li & al. approach was implemented as described in Li et al. (2019). The experimental study was conducted without any special tuning. A large number of experiments have been performed to show the accuracy and the effectiveness of our augmentation-based ensemble learning. Due to the lack of space, only few results are presented herein.

As there are no agreed-upon benchmark datasets for stance and fake news detection (Li et al., 2019), we used two publicly available and complementary datasets: FNC-1 (Slovikovskaya, 2019) and FNN (i.e. FakeNewsNet) (Shu, 2019). FNC was released to explore the task of stance detection in the context of fake news detection. As reported in Table 1, the FNC-1 dataset consists of approximately 50k headline-article pairs in the training set and 25k pairs in the test set. Stance detection is a multinomial classification problem, where the relative stance of each headline-article pair has to be classified as

Table 1. Corpus statistics and class distribution in the FNC-1 dataset.

Dataset	# Instances of training data	# Agree	# Disagree	# Discuss	# Unrelated
FNC-1	49,972	7.36%	1.68%	17.82%	73.31%

Headline: Hundreds of Palestinians flee floods in Gaza as Israel opens dams	
Agree (AGR)	GAZA CITY (Ma'an) – Hundreds of Palestinians were evacuated from their homes Sunday morning after Israeli authorities opened a number of dams near the border, flooding the Gaza Valley in the wake of a recent severe winter storm. The Gaza Ministry of Interior said in a statement that civil defense services and teams from the Ministry of Public Works had evacuated more than 80 families from both sides of the Gaza Valley (Wadi Gaza) after their homes flooded as water levels reached more than three meters [..]
Discuss (DSC)	Palestinian officials say hundreds of Gazans were forced to evacuate after Israel opened the gates of several dams on the border with the Gaza Strip, and flooded at least 80 households. Israel has denied the claim as "entirely false". [..]
Disagree (DSG)	Israel has rejected allegations by government officials in the Gaza strip that authorities were responsible for released storm waters flooding parts of the besieged area. "The claim is entirely false, and southern Israel does not have any dams," said a statement from the Coordinator of Government Activities in the Territories (COGAT). "Due to the recent rain, streams were flooded throughout the region with no connection to actions taken by the State of Israel." At least 80 Palestinian families have been evacuated after water levels in the Gaza Valley (Wadi Gaza) rose to almost three meters. [..]
Unrelated (UNR)	Apple is continuing to experience 'Hairgate' problems but they may just be a publicity stunt [..]

Figure 3. Headline and text snippets with respective stances from the FNC dataset (Slovikovskaya, 2019).

either: *Agree* if the article agrees with the headline claim, *Disagree* if the article disagrees with the claim, *Discuss* if the article is related to the claim, but takes no position on the subject, and *Unrelated* if the content of the article is unrelated to the claim. An illustration of the above classification task is given in Figure 3.

It is worth mentioning that the discovery of a disagreeing headline-article pair does not necessarily correspond to the discovery of a fake article, but is an automated first step which could make human fact-checkers aware of a discrepancy (Momchil et al., 2022). Human fact-checkers or specialized algorithms can then ultimately decide which articles are fake, based on the credibility of agreeing and disagreeing sources.

FNN data was collected from two fact-checking websites (i.e. GossipCop and PolitiFact) containing news contents, along with context information. In comparison with FNN, FNC-1 provides fewer data features (4 vs. 13 features), but more data (≈ 75k vs. ≈ 16k) (Table 2).

6.2. Results and discussion

We ran our experiments with four objectives in mind: (i) identify the best performing (*Augmentation technique*, *Classifier*) pairs; (ii) quantify the actual performance improvement allowed by each text augmentation technique; (iii) evaluate the effectiveness of our augmentation-based ensemble approach; and (iv) evaluate the effectiveness of text augmentation in addressing class imbalance.

Table 2. Class distribution in the FNN dataset (Shu, 2019).

Dataset	# Instances	% True news	% Fake news
FNN	16,118	76.23%	23.77%

6.2.1. Best performing pairs

The two Tables 3 and 5 (resp. Figures 4 and 6), report the F1-scores and Accuracy obtained on FNC (resp. FNN). The results presented in these tables allow to draw important conclusions regarding text augmentation:

(1) *Text augmentation does not always improve predictive performance*. This can be especially observed for SVM, LightGBM, GradBoost (Tables 3 and 5) and AdaBoost

Table 3. F1-score on FNC without and with data augmentation.

Classification algorithm	Without data augmentation	With data augmentation					
		Synonym	BERT	Tf-IDF subtitution	TF-IDF insertion	Back-translation	Combination
DT	81.64%	75.51%	84.01%	80.59%	81.71%	**86.80%**	78.05%
SVM	**61.78%**	61.78%	61.78%	61.78%	61.78%	61.78%	61.78%
AdaBoost	61.71%	**62.08%**	61.86%	61.91%	61.82%	61.78%	61.83%
RF	85.47%	80.22%	88.27%	84.77%	85.55%	**89.93%**	82.95%
XGBoost	75.04%	71.67%	72.65%	71.91%	71.92%	73.91%	71.67%
Bagged RF	85.99%	80.54%	88.25%	85.04%	85.80%	**90.24%**	83.23%
LightGBM	**86.48%**	78.74%	84.07%	82.16%	82.61%	86.20%	80.14%
GradBoost	**72.98%**	70.89%	71.68%	70.97%	70.88%	72.75%	70.87%
LR	62.40%	62.65%	62.39%	62.45%	**62.67%**	62.50%	62.42%
NB	63.54%	62.56%	62.91%	62.71%	63.03%	**63.65%**	63.31%

Table 5. Accuracy on FNC without and with text augmentation.

Classification algorithm	Without data augmentation	With data augmentation					
		Synonym	BERT	TF-IDF substitution	TF-IDF insertion	Back-translation	Combination
DT	81.48%	75.27%	84%	80.34%	81.60%	**86.72**%	77.92%
SVM	**73.13**%	73.13%	73.13%	73.13%	73.13%	73.13%	73.13%
AdaBoost	72.45%	72.59%	72.73%	72.91%	**73.07**%	72.77%	72.84%
RF	86.82%	82.77%	89.09%	86.12%	86.69%	**90.60%**	84.77%
XGBoost	**80.13**%	77.89%	78.47%	78.03%	77.97%	79.26%	77.84%
Bagged RF	87.06%	82.66%	88.99%	86.14%	86.61%	**90.85%**	84.84%
LightGBM	**87.67**%	81.94%	85.77%	84.37%	84.66%	87.36%	82.88%
GradBoost	**78.66%**	77.28%	77.73%	77.38%	77.22%	78.36%	77.26%
LR	73.30%	73.44%	73.33%	73.38%	**73.47%**	73.33%	73.35%
NB	68.81%	68.09%	68.43%	67.99%	68.71%	68.79%	**69.64%**

Table 4. F1-score on FNN without and with data augmentation.

Classification algorithm	Without data augmentation	With data augmentation					
		Synonym	BERT	TF-IDF substitution	TF-IDF insertion	Back-translation	Combination
DT	81.80%	86.18%	86.31%	**87.70%**	87.66%	85.97%	86.30%
SVM	86.78%	86.79%	**86.80%**	86.65%	86.65%	86.78%	86.77%
AdaBoost	**87.92%**	87.83%	87.84%	87.88%	87.89%	87.85%	87.84%
RF	88.31%	90.19%	90.38%	**91.15%**	**91.15%**	90.03%	90.68%
XGBoost	88.15%	88.68%	88.67%	88.72%	**88.79%**	88.74%	88.71%
Bagged RF	88.09%	90.46%	90.40%	91.29%	**91.34%**	90.33%	90.68%
LightGBM	87.84%	89.41%	89.35%	**89.84%**	89.81%	89.29%	89.49%
GradBoost	88.07%	88.23%	88.15%	**88.23%**	88.21%	88.20%	88.18%
LR	86.85%	**87%**	86.98%	86.96%	86.97%	86.92%	**87%**
NB	86.46%	86.57%	86.51%	86.56%	**86.59%**	86.56%	**86.59%**

(Tables 4 and 6), where the F1-scores and Accuracy on the original dataset are higher than to those obtained on the augmented datasets;

(2) *There is no one-size-fits-all augmentation technique that performs well in all situations*. As shown in Tables 3 and 4 (resp. Figures 5 and 6), an augmentation technique may perform well when combined with a classification algorithm and poorly when combined with another. This is the case for example for the 'Synonym' technique which yields the highest F1-score when combined with AdaBoost and the lowest score when used with Naive Bayes (Table 3).

It is worth noting that even if BERT doesn't achieve the highest F1-scores, it provides a steady performance improvement for almost all classifiers;

(3) *The motto 'the more, the better' is the wrong approach regarding text augmentation and targeted approaches allow often better results*. This can be observed in Tables 3 and 4 (resp. Figures 5 and 6), where in almost all cases, combining all augmentation techniques does not yield the best F1-scores and Accuracy.

As shown in Tables 3 and 5, the pairs (*Back-translation, Bagged RF*) and (*Back-translation, RF*) yield the best performance on FNC and increase substantially the predictive performances ($\approx +4.16\%$ in comparison with the highest F1-Score that can be achieved without text augmentation). Similarly, as shown in Tables 4 and 6, the pairs (*Substitution TF-IDF, RF*) and (*Insertion TF-IDF, Bagged RF*) yield the best performance on the dataset FNN ($\approx +5.87\%$).

6.2.2. Augmentation-based ensemble learning

As previously stated, base learners' diversity and competency are the two key success factors of any ensemble learning approach. Our ensemble approach leverages text augmentation to enhance both. Figure 2 depicts our classification model which is a mixture of stacking and bagging. In our model, we use Bagged RF and Random Forest (RF) as base classifiers and GradBoost as meta-classifier. As depicted in Figure 2, each of the base classifiers is trained on a dataset composed of the original dataset and the data obtained by applying one of the augmentation techniques. The choice of the (classifier, augmentation technique) pairs was driven by the experimental study conducted in Subsection 6.2.1. We

Table 6. Accuracy on FNN without and with data augmentation.

Classification algorithm	Without data augmentation	With data augmentation					
		Synonym	BERT	TF-IDF substitution	TF-IDF insertion	Back-translation	Combination
DT	72.66%	78.98%	79.04%	81.10%	**81.34%**	78.60%	79.17%
SVM	76.83%	76.86%	**76.88%**	76.53%	76.53%	76.84%	76.81%
AdaBoost	**79.95%**	79.77%	79.77%	79.86%	79.87%	79.87%	79.79%
RF	81.02%	84.14%	84.42%	**85.75%**	85.72%	83.87%	84.94%
XGBoost	80.38%	81.27%	81.27%	81.36%	**81.50%**	81.39%	81.33%
Bagged RF	80.75%	84.48%	84.59%	86.06%	**86.07%**	84.36%	85.10%
LightGBM	80.30%	82.80%	82.70%	**83.57%**	83.49%	82.64%	82.97%
GradBoost	80.23%	80.46%	80.30%	80.45%	80.41%	**80.49%**	80.36%
LR	77.26%	**77.63%**	77.59%	77.52%	77.56%	77.52%	77.63%
NB	77.06%	**77.24%**	77.14%	77.17%	77.23%	77.22%	77.23%

Table 7. F1-score and accuracy achieved by conventional stacking, Li et al. (2019) and the proposed approach.

Model	F1-score		Accuracy	
	FNC	FNN	FNC	FNN
(TF-IDF Insertion. Stacking)	85.58%	**90.92%**	86.62%	**85.67%**
(TF-IDF Substitution. Stacking)	84.57%	90.43%	85.78%	84.72%
(Back-Translation. Stacking)	**90.31%**	89.80%	**89.70%**	83.47%
(BERT. Stacking)	87.93%	90.26%	88.62%	84.23%
(Synonym. Stacking)	80.71%	90.28%	82.71%	84.17%
(Combination. Stacking)	83.11%	90.73%	84.98%	85.28%
Li et al. (2019)	83.72%	88.45%	83.67%	79.31%
Proposed approach	**90.15%**	**91.07%**	**90.67%**	**85.78%**

compare our model to a more classical stacking approach, where all base classifiers are trained on the same dataset, consisting of the original dataset and the data obtained by applying one of the augmentation techniques (Figure 1). We also compare our model to the approach of Li et al. (2019), which is one of the state-of-the-art approaches that uses LSA, stacking-based ensemble learning and K-fold cross-validation. Table 7 synthesizes the predictive performances achieved by each approach.

As reported in Table 7, the use of text augmentation allows better performances than those achieved by Li et al. (2019) in almost all situations. On the other hand, except for the Synonym technique over the FNC dataset, our model outperforms the classical approach in all situations. Overall, our stacking approach achieves an increase in F1-score and Accuracy of 7,72% and 7,13% (resp. 7,54% and 2,88%) over FNC (resp. FNN) when compared to Li et al. (2019).

6.2.3. Class balancing

The class imbalance problem arises when data is distributed unevenly among classes; i.e. when one or more of the predicted outputs happen much less frequently than others. As stated in Fernndez et al. (2018), when working with an imbalanced classification problem:

- *The minority classes are typically of the most interest*, meaning that a model's skill in correctly predicting a minority class is more important than in correctly predicting a majority class.
- *Minority classes are harder to predict*. The main reason is that with few available training examples, it is often challenging to identify regularities and learn characteristics. That is why most classification algorithms tend to be biased towards the majority class(es), causing bad classification of the minority class(es).

The above two observations hold for the tasks of stance and fake news detection. As stated by the authors of the FNC challenge (Pomerleau & Rao, 2017),

> *'The related/unrelated classification task is expected to be much easier and is less relevant for detecting fake news…. The Stance Detection task (classify as agrees, disagrees or discuss) is both more difficult and more relevant to fake news detection, …'.*

When dealing with class imbalance, evaluation metrics that give equal importance to each observation (and not to each class), such as the *Accuracy*, can be misleading as they

Table 8. Per class F1-score on balanced data (FNN).

Model	Original dataset		Balanced FNN	
	False	True	False	True
DT	44.90%	82.43%	**85.18%**	**84.47%**
SVM	25.22%	87.93%	**66.26%**	74.16%
AdaBoost	40.42%	88.49%	**69.54%**	73.67%
RF	49.59%	88.35%	**88.21%**	88.02%
XGBoost	42.46%	88.74%	**69.90%**	74.58%
Bagged RF	48.04%	86.20%	**86.79%**	85.42%
LightGBM	43.01%	88.07%	**72.15%**	76.09%
GradBoost	42.15%	88.69%	**70.40%**	74.81%
LR	14.82%	87.07%	**66.89%**	72.50%
NB	24.37%	86.77%	**53.92%**	72.48%
Proposed approach	46.11%	88.84%	**87.14%**	87.65%

fail to reflect the poor performance over the minority classes (*Accuracy Paradox*). In the sequel, in order to better perceive the effects of class imbalance we provide per class F1-scores.

The FNN dataset is moderately imbalanced and contains 12287 *True News* and only 3831 *Fake News*. The FNC dataset has a more severe imbalance as it contains 36 545 *Unrelated*, 8 909 *Discuss*, and only 3 678 *Agree*, and 840 *Disagree* examples. The bias towards the majority classes can be observed in Tables 8 and 9 where all algorithms perform poorly over the minority classes *Fake News* (Table 8) and *Agree/Diagree* (Table 9). Some algorithms, such as SVM, Logistic Regression and Naïve Baies even obtain zero F1-scores over the minority classes *Agree/Disagree*.

To balance the FNN dataset, 6765 additional samples was generated for the minority class *Fake News*, using the five augmentation techniques of section 3. Similarly, we generated 18 390 additional examples for the *Agree* class and 4 200 additional examples for the *Disagree* class. We should notice here that the number of examples of the *Diagree* class remains much less than the number of instances of the majority class *Unrelated*. As shown in Tables 8 and 9, text augmentation allows a substantial improvement in the F1-scores obtained over the minority classes: an average improvement of 94.47% for the *Fake News* class (Table 8), 189.14% for the *Agree* class and 586.39% for the *Disagree* class (Table 9).

Table 9. Per class F1-score on balanced data (FNC).

Model	Original dataset				Balanced FNC			
	Agree	Disagree	Discuss	Unrelated	Agree	Disagree	Discuss	Unrelated
DT	40.70%	28.49%	68.80%	89.95%	**69.47%**	**84.52%**	59.06%	81.13%
SVM	0%	0%	33.51%	81.36%	**36%**	**38.89%**	21.48%	63.27%
AdaBoost	0%	0%	0.77%	84.30%	**31.26%**	**42.30%**	6.08%	59.09%
RF	45.99%	27%	74.74%	92.53%	**78.11%**	**89.53%**	68.08%	87.11%
XGBoost	1.85%	0%	41.08%	86.98%	**43.34%**	**45.28%**	34.10%	66.14%
Bagged RF	45.68%	28.87%	72.61%	91.94%	**74.56%**	**87.25%**	66.09%	84.77%
LightGBM	14.65%	8.56%	57.81%	88.99%	**54.40%**	**74.54%**	47.30%	75.13%
GradBoost	6.17%	3.45%	46.96%	87.56%	**44.87%**	**54.85%**	39.81%	68.12%
LR	0%	0%	2.42%	84.50%	**1.01%**	**28.20%**	2.86%	59.04%
NB	0%	0%	24.70%	80.92%	**15.27%**	**28.87%**	18.83%	57.75%
Proposed approach	47.78%	26.34%	75.35%	92.64%	**77.23%**	**89.26%**	65.85%	86.82%

7. Conclusion

Combating fake news on social media is a pressing need and a daunting task. Most of existing approaches on fake news detection, focus on using various features to identify those allowing the best predictive performance. Such approaches tend to undermine the generalization ability of the obtained models.

In this work, we investigated the use of text augmentation in the context of stance and fake news detection. In the first part of our work, we studied the effect of text augmentation on the performance of various classification algorithms. Our experimental study quantified the actual contribution of data augmentation and identified the best performing (*classifier, augmentation technique*) pairs. Besides, our study revealed that the motto 'the more, the better' is the wrong approach regarding text augmentation and that there is no one-size-fits-all augmentation technique. In the second part of our work, we proposed a novel augmentation-based ensemble learning approach. The proposed approach is a mixture of bagging and stacking and leverages text augmentation to enhance the diversity and the performance of base classifiers. We evaluated our approach using two real-world datasets. Experimental results show that the proposed approach is more accurate than state-of-art methods. In the third part of our work we investigated the use of text augmentation to cope with class imbalance, a very common problem in stance and fake news detection. As shown by our experimental study, even in presence of severe imbalance, text augmentation can highly alleviate its effects and substantially improve the predictive performance over the minority classes.

As a part of our future work, we intend to explore the use of a multimodal data augmentation that involves linguistic and extra linguistic features.We also intend to explore the detection of fake news from streams under concept drifts and to connect the dots between stance detection and fake news detection through the use of media profiling and multi-source credibility scores.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Notes on Contributors

Ilhem Salah is currently pursuing a Ph.D. in Computer Science at Sousse University (Tunisia) after earning her Master's degree in Distributed Computing from the same institution. Her research interests include Machine Learning, Distributed Ledgers, and NLP.

Khaled Jouini received the Ph.D. degree in Computer Science from Paris-Dauphine University (France) in 2008. He was a research staff member at Telecom ParisTech (France). Since 2011, he has been with Sousse University (Tunisia), where he is currently an Associate Professor. His research interests include Data Engineering, Machine Learning, NLP, and Large-scale data management and mining.

Ouajdi Korbaa is a full-time professor at the University of Sousse (Tunisia). He received his Engineering Diploma from the Ecole Centrale de Lille (France) in 1995 and his Master's degree in Production Engineering and Computer Science from the University of Lille (France) in the same year. He obtained his Ph.D. in Production Management, Automatic Control, and Computer Science from the University of Science and Technologies of Lille (France) in 1998 and his "Habilitation to Supervise Researches" degree in Computer Science from the same University in 2003. Pr. Korbaa

has published around 150 research papers on Optimistation, Simultation and Modeling, Applied and Computational Mathematics, Manufacturing Engineering and Computer Engineering.

ORCID

Ilhem Salah http://orcid.org//0000-0002-3375-3637
Khaled Jouini http://orcid.org//0000-0001-5049-4238
Ouajdi Korbaa http://orcid.org/0000-0003-4462-1805

References

Ahmed, H., Traore, I., & Saad, S. (2017). Detection of online fake news using n-gram analysis and machine learning techniques. In *Intelligent, secure, and dependable systems in distributed and cloud environments: First international conference* (pp. 127–138). Springer.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, *3*(2003), 993–1022.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, *41*(6), 391–407.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. CoRR, abs/1810.04805

Dulhanty, C., Deglint, J. L., Ben Daya, I., & Wong, A. (2019). Taking a stance on fake news: Towards automatic disinformation assessment via deep bidirectional transformer language models for stance detection. CoRR, abs/1911.11951

Fernndez, A., Garca, S., Galar, M., Prati, R. C., Krawczyk, B., & Herrera, F. (2018). *Learning from imbalanced data sets* (1st ed.). Springer Publishing Company, Incorporated.

Guo, Z., Schlichtkrull, M. S., & Vlachos, A. (2021). A survey on automated fact-checking. CoRR, https://arxiv.org/abs/2108.11896

Hanselowski, A., PVS, A., Schiller, B., Caspelherr, F., Chaudhuri, D., Meyer, C. M., & Gurevych, I. (2018). A retrospective analysis of the fake news challenge stance-detection task. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018)*. Association for Computational Linguistics.

Jiang, T., Li, J. P., Haq, A. U., Saboor, A., & Ali, A. (2021). A novel stacking approach for accurate detection of fake news. *IEEE Access*, *9*(2021), 22626–22639.

Jouini, K., Maaloul, M. H., & Korbaa, O. (2021). Real-time CNN-based assistive device for visually impaired people. In *2021 14th International Congress on Image and Signal Processing, Biomedical Engineering and Informatics (CISP-BMEI)* (pp. 1–6). IEEE.

Kaggle.com (n.d.). *Fake and real news dataset*. Retrieved February 19, 2023, from https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset/discussion

Karnyoto, A. S., Sun, C., Liu, B., & Wang, X. (2022). Augmentation and heterogeneous graph neural network for AAAI2021-COVID-19 fake news detection. *International Journal of Machine Learning and Cybernetics*, *13*(7), 2033–2043. https://doi.org/10.1007/s13042-021-01503-5

Khan, J. Y., Khondaker, M. T. I., Afroz, S., Uddin, G., & Iqbal, A. (2021). A benchmark study of machine learning models for online fake news detection. *Machine Learning with Applications*, *4*(June), 2666–8270. https://doi.org/10.1016/j.mlwa.2021.100032

Li, B., Hou, Y., & Che, W. (2021). Data augmentation approaches in natural language processing: A survey. CoRR, abs/2110.01852

Li, S., Ma, K., Niu, X., Wang, Y., Ji, K., Yu, Z., & Chen, Z. (2019). Stacking-based ensemble learning on low dimensional features for fake news detection. In *2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)* (pp. 2730–2735). IEEE.

Ma, E. (2019). *NLP augmentation*. Retrieved May 15, 2021, from https://github.com/makcedward/nlpaug

Mahabub, A. (2020). A robust technique of fake news detection using ensemble voting classifier and comparison with other classifiers. *SN Applied Sciences*, *2*(4), 525. https://doi.org/10.1007/s42452-020-2326-y

Marivate, V., & Sefara, T. (2019). Improving short text classification through global augmentation methods. CoRR, abs/1907.03752

McIntire, G. (2017). *Machine learning finds 'fake news' with 88% accuracy*. Kdnuggets, ODSC. Retrieved February 19, 2023, from https://www.kdnuggets.com/2017/04/machine-learning-fake-news-accuracy.html

Momchil, H., Arnav, A., Preslav, N., & Isabelle, A. (2022). A survey on stance detection for mis- and disinformation identification. In *Findings of the Association for Computational Linguistics: NAACL 2022* (pp. 1259–1277). Association for Computational Linguistics.

NLTK.org. (n.d.). *Natural language toolkit*. Retrieved May 15, 2021, from https://github.com/nltk/nltk

Pan, Y. (2018). *Fake news challenge – team solat in the swen*. Retrieved February 22, 2023, from https://github.com/Cisco-Talos/fnc-1/

Patil, D. R. (2022). Fake news detection using majority voting technique. arXiv, https://arxiv.org/abs/2203.09936

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*(2011), 2825–2830. https://doi.org/10.48550/arXiv.1201.0490

Pomerleau, D., & Rao, D. (2017). *The fake news challenge: Exploring how artificial intelligence technologies could be leveraged to combat fake news*. Retrieved April 1, 2022, from https://www.fakenewschallenge.org/

Riedel, B., Augenstein, I., Spithourakis, G. P., & Riedel, S. (2017). A simple but tough-to-beat baseline for the fake news challenge stance detection task. CoRR, http://arxiv.org/abs/1707.03264

Salah, I., Jouini, K., & Korbaa, O. (2022). Augmentation-based ensemble learning for stance and fake news detection. In *Advances in Computational Collective Intelligence – 14th International Conference, ICCCI 2022, Proceedings of Communications in Computer and Information Science* (Vol. 1653, pp. 29–41). Springer.

Sepúlveda-Torres, R., Vicente, M., Saquete, E., Lloret, E., & Palomar, S. M. (2021, November). HeadlineStanceChecker: Exploiting summarization to detect headline disinformation. *Journal of Web Semantics*, *71*, 100660 https://doi.org/10.1016/j.websem.2021.100660

Shi, L., Liu, D., Liu, G., & Meng, K. (2020). AUG-BERT: An efficient data augmentation algorithm for text classification. In *Communications, signal processing, and systems* (pp. 2191–2198). Springer.

Shoemaker, E. (2019). *Using data science to detect fake news*. James Madison University JMU Scholarly Commons, https://orcid.org/0000-0002-7955-5441

Shorten, C., Khoshgoftaar, T., & Furht, B. (2021). Text data augmentation for deep learning. *Journal of Big Data*, *8*(1), 1–34. https://doi.org/10.1186/s40537-021-00492-0

Shu, K. (2019). FakeNewsNet. Retrieved December 15, 2021, from https://doi.org/10.7910/DVN/UEMMHS, Harvard Dataverse, V2

Slovikovskaya, V. (2019). Transfer Learning from Transformers to Fake News Challenge Stance Detection {(FNC-1)} Task. *Proceedings of the Twelfth Language Resources and Evaluation Conference*. European Language Resources Association.

Surowiecki, J. (2005). *The wisdom of crowds* (1st ed.). Anchor Books.

Suting, Y., & Ning, Z. (2020). Construction of structural diversity of ensemble learning based on classification coding. In *2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)* (Vol. 9, pp. 1205–1208). *IEEE*.

Tesfagergish, S. G., Damaševičius, R., & Kapočiūtė-Dzikienė, J. (2021). Deep fake recognition in tweets using text augmentation, word embeddings and deep learning. In *ICCSA 2021: 21st International Conference Computational Science and Its Applications* (pp. 523–538). Springer Nature.

Wang, W. Y. (2017). "Liar, liar pants on fire": A new Benchmark dataset for fake news detection. CoRR, abs/1705.00648

Xie, Q., Dai, Z., Hovy, E. H., Luong, M. T., & Le, Q. V. (2019). Unsupervised data augmentation. CoRR, abs/1904.12848

.

Q2.2 Intrusion Detection based on Concept Drift Detection & Online Incremental Learning

Farah Jemili, Khaled Jouini & Ouajdi Korbaa

International Journal of Pervasive Computing and Communications (INT J PERVASIVE COMP).
Vol. 21 No. 1, pp. 81-115. 2025.
ISSN : 1742-7371 , Emerald group publishing ltd,
JCR IF : 0.6

DOI : https://doi.org/10.1108/IJPCC-12-2023-0358
SJR best quartile : Q2, SJR : 0.36

International Journal of Pervasive Computing and Communications

COUNTRY

United Kingdom

Universities and research institutions in United Kingdom

Media Ranking in United Kingdom

SUBJECT AREA AND CATEGORY

Computer Science
Computer Science (miscellaneous)

Mathematics
Theoretical Computer Science

PUBLISHER

Emerald Group Publishing Ltd.

H-INDEX

23

PUBLICATION TYPE

Journals

ISSN

17427371, 1742738X

COVERAGE

2005, 2007-2023

INFORMATION

Homepage

How to publish in this journal

George.Ghinea@brunel.ac.uk

SCOPE

The International Journal of Pervasive Computing and Communications (IJPCC) is multi-disciplinary and inter-disciplinary vehicle to discuss the future where computers and computing devices will be available naturally and unobtrusively everywhere, anytime, and by different means in our daily living, working, learning, business, infotainment environments. Tremendous opportunities exist for novel services/applications that are more immersive, more intelligent, and more interactive in both real and cyber spaces. IJPCC is thus a premier channel to share research in the emerging field of pervasive computing and communications including future directions and issues. Its published research includes: experimental or theoretical results, novel algorithms, design methodologies, work-in-progress, experiences, case studies, and trend-setting ideas.

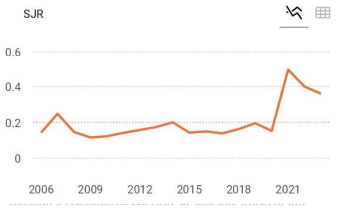
Join the conversation about this journal

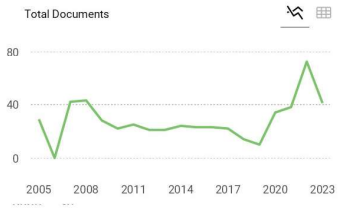
Quartiles

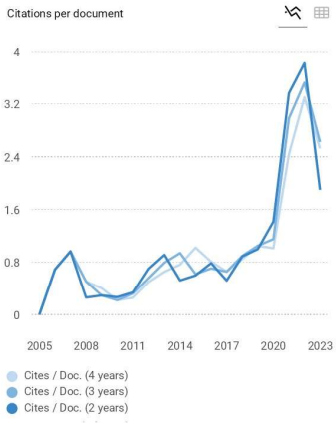
FIND SIMILAR JOURNALS

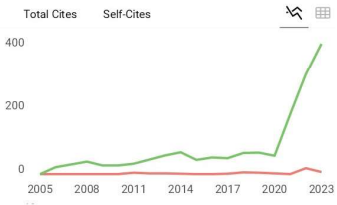
options

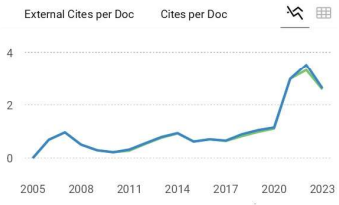
1	2	3	4	5
Wireless Communications and Mobile Computing	**Scalable Computing**	**International Journal of Ad Hoc and Ubiquitous**	**Security and Communication Networks**	**Egyptian Informatics Journal**
GBR	ROU	GBR	GBR	NLD
74%	71%	70%	70%	70%

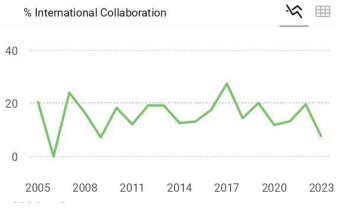


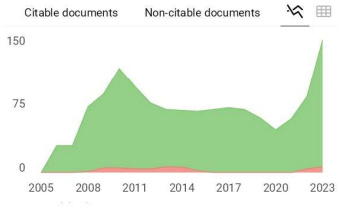


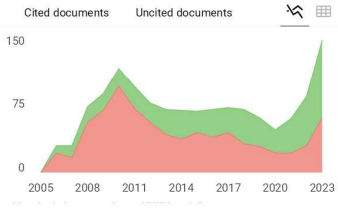


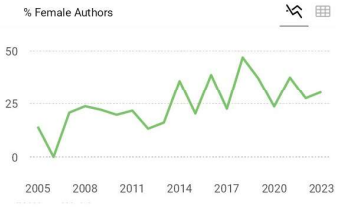


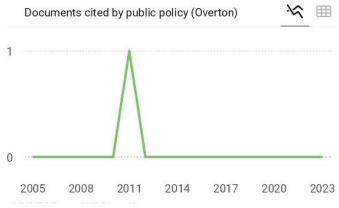


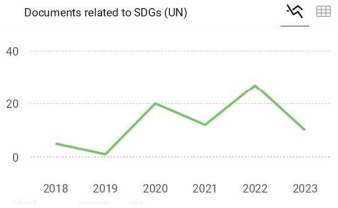












← Show this widget in your own website

Just copy the code below and paste within your html code:

<a href="https://www.scim

SCImago Graphica

Explore, visually communicate and make sense of data with our **new data visualization tool**.

Intrusion Detection based on Concept Drift Detection & Online Incremental Learning

Journal:	*International Journal of Pervasive Computing and Communications*
Manuscript ID	IJPCC-12-2023-0358.R1
Manuscript Type:	Research Paper
Keywords:	Concept Drift Detection, Online Incremental learning, Online Random Forest, Intrusion Detection, Structured Streaming

MANUSCRIPT DETAILS

TITLE: Intrusion Detection based on Concept Drift Detection & Online Incremental Learning

ABSTRACT:

The primary purpose of this paper is to introduce the DDM-ORF model for intrusion detection, combining Drift Detection Method (DDM) for detecting concept drift and Online Random Forest (ORF) for incremental learning. The paper addresses the challenges of dynamic and non-stationary data, offering a solution that continuously adapts to changes in the data distribution. The goal is to provide effective intrusion detection in real-world scenarios, demonstrated through comprehensive experiments and evaluations using Apache Spark.

The paper employs an experimental approach to evaluate the DDM-ORF model. The design involves assessing classification performance metrics, including accuracy, precision, recall, and F-measure. The methodology integrates Apache Spark for distributed computing, utilizing metrics such as Processed Records per Second and Input Rows per Second. The evaluation extends to the analysis of IP addresses, ports, and taxonomies in the MAWILab dataset. This comprehensive design and methodology showcase the model's effectiveness in detecting intrusions through concept drift detection and online incremental learning on large-scale, heterogeneous data.

The paper's findings reveal that the DDM-ORF model achieves outstanding classification results with 99.96% accuracy, demonstrating its efficacy in intrusion detection. Comparative analysis against a CNN-based model indicates superior performance in anomalous and suspicious detection rates. The exploration of IP addresses, ports, and taxonomies uncovers valuable insights into attack patterns. Apache Spark evaluation attests to the system's high processing rates. The study emphasizes the scalability, availability, and fault-tolerance of DDM-ORF, making it suitable for real-world scenarios. Overall, the paper establishes the model's proficiency in handling dynamic, non-stationary data for intrusion detection.

The research acknowledges certain limitations, including the potential challenge of DDM detecting only frequency changes in class labels and not complex concept drifts. The incremental random forest's reliance on memory may pose constraints as the forest size increases, potentially leading to overfitting. Addressing these limitations could involve exploring alternative concept drift detection algorithms and implementing ensemble pruning techniques for memory efficiency. Further research avenues may investigate algorithms balancing accuracy and memory usage, such as compressed random forests, to enhance the model's effectiveness in evolving data environments.

The study's practical implications are noteworthy. The proposed DDM-ORF model, designed for intrusion detection through concept drift detection and online incremental learning, offers a scalable, available, and fault-tolerant solution. Leveraging Apache Spark and Microsoft Azure Cloud enhances processing capabilities for large datasets in dynamic, non-stationary scenarios. The model's applicability to heterogeneous datasets and its achievement of high-accuracy multi-class classification make it suitable for real-world intrusion detection. Moreover, the auto-scaling features of Microsoft Azure Cloud contribute to adaptability, ensuring efficient resource utilization without

downtime. These practical implications underscore the model's relevance and effectiveness in diverse operational contexts.

The DDM-ORF model's social implications are significant, contributing to enhanced cybersecurity measures. By providing an effective intrusion detection system, it helps safeguard digital ecosystems, preserving user privacy and securing sensitive information. The model's accuracy in identifying and classifying various intrusion attempts aids in mitigating potential cyber threats, thereby fostering a safer online environment for individuals and organizations. As cybersecurity is paramount in the digital age, the social impact lies in fortifying the resilience of networks, systems, and data against malicious activities, ultimately promoting trust and reliability in online interactions.

The DDM-ORF model introduces a novel approach to intrusion detection by combining drift detection and online incremental learning. This originality lies in its utilization of the Drift Detection Method and Online Random Forest algorithm, offering a dynamic and adaptive system for evolving data. The model's contribution extends to its scalability, fault-tolerance, and suitability for heterogeneous datasets, addressing challenges in dynamic, non-stationary environments. Its application on a large-scale dataset and multi-class classification, along with integration with Apache Spark and Microsoft Azure Cloud, enhances the field's understanding and application of intrusion detection, providing valuable insights for securing digital infrastructures.

Intrusion Detection based on Concept Drift Detection & Online Incremental Learning

Abstract- Intrusions are constantly evolving and changing, and to keep up with these changes, it is necessary to have models that detect these changes, also known as concept drifts, and offer the ability to update the model without starting the learning process from scratch. In our contribution, we introduce a novel approach to intrusion detection that leverages both concept drift detection and online incremental learning, named DDM-ORF. While traditional IDS methods struggle to adapt to the evolving nature of cyber threats, our approach uniquely integrates the Detection Drift Method (DDM) with the Online Random Forest (ORF) algorithm, providing real-time adaptability and high accuracy. Unlike existing methods that rely on batch processing and are limited to binary classification, DDM-ORF offers multi-class classification and continuous learning capabilities, making it exceptionally suited for handling massive and dynamic data streams in real-world applications. This innovative combination addresses the critical need for scalability and adaptability in IDS. The proposed system achieves very good classification results, along with good processing speed that meets real-world scenarios. Apache Spark Structured Streaming provides important functionalities for dealing with streaming data and enables the deployment of the proposed system DDM-ORF in real-world applications.

Keywords: Concept Drift Detection, Online Incremental learning, Online Random Forest, Intrusion Detection, Structured Streaming.

1. INTRODUCTION

In today's complex landscape of computer networks, the significance of Intrusion Detection Systems (IDS) cannot be overstated—they play a pivotal role in identifying and mitigating potential threats to network security [6]. With the escalating frequency and sophistication of cyber-attacks, IDS has become indispensable for network administrators striving to uphold the confidentiality, integrity, and availability of their systems. Operating on the analysis of network traffic, IDS discerns patterns indicative of potential security breaches. Upon detecting an intrusion, the IDS promptly alerts network administrators, empowering them to take necessary actions to counteract the threat [7].

Yet, despite their crucial role, IDS grapple with several challenges due to the continually evolving nature of attacks and the emergence of new attack patterns collectively termed as concept drift. Traditional IDS find it challenging to adapt to these changes, necessitating frequent manual updates that prove to be time-consuming and costly [8]. For instance, an IDS trained to identify a specific type of attack may lose effectiveness if attackers alter their tactics or employ new methods. In such scenarios, the IDS demands updates with new rules or models to stay abreast of the evolving threat landscape [9].

To counter this challenge, a novel approach grounded in incremental learning and concept drift detection has been proposed. Concept drift, a phenomenon where the statistical properties of the analyzed data change over time, poses a challenge to maintaining IDS accuracy. Incremental learning, a machine learning technique, addresses this by dynamically updating the model as new data is introduced. The integration of concept drift detection and incremental learning empowers an IDS to continuously learn from new data, adapting to shifts in network behavior and effectively identifying previously unknown threats [10]. Enter our proposed approach, DDM-ORF, designed to markedly enhance IDS accuracy and efficacy—an invaluable tool in the ongoing battle against cyber attacks.

DDM-ORF is crafted to be adaptive, detecting concept drifts and seamlessly updating the model with incoming data to uphold accuracy. This approach boasts numerous advantages over traditional IDS, including heightened accuracy, scalability, and the ability to navigate dynamic shifts in data. Leveraging online incremental learning, DDM-ORF updates the model in real-time, proving particularly effective in detecting new and emerging threats. Moreover, with concept drift detection, this approach can discern alterations in data distribution, adjusting the model accordingly. Given the ever-evolving nature of cyber threats, possessing IDS capable of keeping pace with the changing threat landscape is imperative, and DDM-ORF presents a promising solution to this critical issue.

The paper unfolds in six sections: commencing with the Introduction, the second section delves into a review of related works on IDS. The third section elucidates the concept drift detection and online incremental learning techniques underpinning the proposed approach. Following that, the fourth section meticulously details the proposed approach itself. The fifth section outlines the experimental setup employed to evaluate the proposed approach. Finally, the last section concludes the paper, emphasizing its contributions and suggesting future research directions.

2. RELATED WORK

In [1], the authors introduce an innovative intrusion detection approach that integrates ensemble incremental learning to grapple with the challenge of concept drift. Acknowledging the need for Intrusion Detection Systems (IDS) to adapt to new attack patterns and address concept drift, the authors propose an ensemble incremental learning approach. This involves deploying a set of classifiers that incrementally learn from new data while updating their knowledge of previously learned attack patterns. Central to this approach is a concept drift detection module that vigilantly monitors the incoming data stream for shifts in the underlying concept. Upon detecting a concept drift, the system triggers an incremental learning process to align the ensemble of classifiers with new attack patterns. The authors introduce a weighted voting scheme, dynamically adjusting classifier weights based on historical performance and concept drift significance. Results showcase the superior accuracy and adaptability of the proposed ensemble incremental learning approach over traditional IDS methods. However, the reliance on batch learning limits its efficacy in handling evolving data streams, and the binary classification scope may be constraining in scenarios requiring more nuanced classification.

In [2], the focus shifts to developing an IDS tailored for Internet of Things (IoT) environments. Addressing the challenge of securing interconnected IoT networks, the authors propose an adaptive class incremental learning-based IDS. This system incrementally learns and adapts to new attack classes while maintaining high detection accuracy. Employing a feature extraction module to capture relevant characteristics from network traffic data, the IDS integrates a class incremental learning algorithm that combines deep neural networks with an incremental learning approach. This mechanism facilitates learning new attack classes without forgetting previously acquired patterns. An adaptive decision-making component dynamically adjusts detection thresholds based on network conditions, enhancing adaptability and reducing false positives or negatives. While outperforming traditional IDS methods, a potential limitation lies in the initial need for a substantial amount of labeled data and possible performance issues on datasets with significant class imbalance.

In [3], a focus on Industrial Internet of Things (IIoT) environments prompts the authors to propose an intrusion detection method rooted in active incremental learning. Aimed at improving accuracy and efficiency in IIoT systems, the method incorporates active learning where human experts interact with the system, providing feedback and labeling samples. An incremental learning algorithm facilitates continuous updates to the system's knowledge, adapting to evolving attack

patterns. Additionally, a feature selection process enhances efficiency by identifying relevant features. Results exhibit superiority in detection accuracy and efficiency over traditional approaches, yet computational resource demands for the incremental learning algorithm and the focus on a specific type of cyber-attack pose challenges.

In [4], the authors present a robust drift detection method, "Learn to adapt," addressing the challenge of detecting and adapting to concept drift in security-related datasets. Utilizing an ensemble of classifiers trained on different dataset partitions, the method employs an adaptation algorithm to dynamically adjust weights based on individual classifier performance. Results demonstrate superior detection accuracy and adaptability. However, the approach's offline nature, requiring historical data for drift detection, may limit real-time applicability.

In [5], a focus on incremental learning for intrusion detection introduces a method based on a dynamic ensemble of Relevance Vector Machines (RVMs). The incremental learning strategy updates knowledge with newly labeled samples, and the dynamic ensemble mechanism adjusts composition based on individual RVM performance. While outperforming traditional approaches, RVMs' need for sufficient labeled data and their potential black-box nature pose challenges.

In [37], authors present an online deep learning approach for intrusion detection in IoT environments, titled "Intrusion detection in the IoT under data and concept drifts: Online deep learning approach" published in the IEEE Internet of Things Journal. Their method continuously updates a deep neural network with new data to adapt to evolving attack patterns, achieving high detection accuracy and robustness against concept drift. However, the approach requires substantial computational resources and sophisticated deep learning frameworks, which may not be readily available in all IoT deployment scenarios. Additionally, it primarily focuses on binary classification, limiting its scope in identifying and distinguishing between multiple types of attacks.

Numerous approaches have been proposed to address the challenge of concept drift in intrusion detection systems. For instance, [1] presents an ensemble incremental learning approach that adapts to new attack patterns. However, this method relies on batch learning, limiting its efficacy in real-time scenarios. Similarly, [2] focuses on class incremental learning for IoT environments, yet requires substantial labeled data for initial training, posing challenges for immediate deployment in diverse settings. In [5], the use of a dynamic ensemble of Relevance Vector Machines (RVMs) for incremental learning also demonstrates improvements but faces challenges due to the need for sufficient labeled data and the potential black-box nature of RVMs. Methods such as [3] and [4] are limited to binary classification, restricting their applicability in more complex scenarios. Recent works like [37] have explored deep learning approaches for handling concept drift, but these require considerable computational resources and sophisticated frameworks, which may not be readily available in all deployment scenarios.

In contrast, our proposed DDM-ORF method stands out by combining DDM's real-time drift detection with ORF's continuous learning capabilities, ensuring that the model adapts instantaneously to new data without the need for batch processing. Furthermore, DDM-ORF's ability to perform multi-class classification effectively addresses the limitations seen in other methods. Our approach also leverages Apache Spark Structured Streaming, enhancing its scalability and processing speed for handling large-scale, heterogeneous data. These features make our DDM-ORF method not only more effective in diverse and dynamic environments but also more practical for immediate deployment compared to other existing methods.

Table 1 summarizes the key differences between our approach and existing methods, underscoring the unique advantages of DDM-ORF in terms of adaptability, scalability, and real-time applicability.

Aspect/Feature	[1] Ensemble Incremental Learning	[2] Adaptive Class Incremental Learning	[3] Active Incremental Learning	[4] Learn to Adapt Drift Detection	[5] Dynamic Ensemble of RVMs	[37] Deep Learning	DDM-ORF (Our Approach)
Learning Approach	Ensemble Incremental Learning	Adaptive Class Incremental Learning	Active Incremental Learning	Ensemble Learning for Drift Detection	Dynamic Ensemble of RVMs	Deep Learning Model (DNN)	Online Incremental Learning with DDM
Detection Accuracy	Superior to Traditional IDS Methods	Outperforms Traditional IDS	Outperforms Traditional IDS	Superior Detection Accuracy	Outperforms Traditional IDS	High Detection Accuracy	Superior Accuracy
Dataset Size Sensitivity	Not explicitly mentioned	Requires Substantial Labeled Data	Active Learning May Require Expert Interaction	Ensemble Trained on Partitions of Dataset	Requires Sufficient Labeled Data	Requires significant amounts of labeled data	Handles Massive Data
Classification Scope	Binary	Binary	Not Specified	Binary	Multiclass (Dynamic Ensemble)	Binary	Multiclass
Real-Time Applicability	Limited due to Batch Learning	Real-Time Applicability Not Specified	Real-Time Applicability Not Specified	Offline Detection Requires Historical Data	Real-Time Applicability Not Specified	Real-Time Detection	Real-Time Detection
Computational Resources	Not specified	Potential Performance Issues with Imbalanced Datasets	Significant Resources for Incremental Learning	Drift Detection Requires Computational Resources	Sufficient Labeled Data for RVMs	Requires substantial computational resources	Low Computational Overhead

Table 1: Related Works

By integrating DDM with ORF and leveraging Apache Spark Structured Streaming, our DDM-ORF approach addresses critical limitations of existing IDS methods, offering a scalable, real-time solution capable of adapting to the continuously evolving threat landscape. This combination of techniques presents a significant advancement in the field of intrusion detection, providing a robust framework for future research and development in cybersecurity.

3. CONCEPT DRIFT DETECTION AND INCREMENTAL LEARNING

In this section, we describe the concept drift detection and online incremental learning techniques that are used in the literature, in order to make appropriate choices for our contribution.

3.1 Concept Drift Detection

Concept drift occurs when the target changes over a limited period of time. Consider two target concepts, A and B, and a sequence of samples, $I = i_1, i_2, ..., i_n$. Prior to a certain instance, the target concept remains unchanged in A. However, after the instance, a new concept, Δx, becomes stable and replaces A with B. The speed of this transition can be gradual or abrupt depending on the efficiency of the drift, Δx. There are three different ways to model concept drift: window-related,

weight-related, and an ensemble of classification models. The former selects samples from a sliding window, while the latter weights samples and removes them based on their weight [7].

There are two approaches to handling concept drift: online and batch. The former updates the classifier after each instance, while the latter waits to receive massive instances before starting the learning process.

There are several concept drift detection techniques that are commonly used in machine learning, including [10] :

- The KS (Kolmogorov-Smirnov) test compares the distribution of the incoming data to a reference distribution and detects significant deviations that may indicate a drift. The KS test can be effective for detecting sudden and significant changes in the data distribution but may be less effective for detecting gradual or subtle changes. The KS statistic measures the maximum distance between the cumulative distribution functions (CDFs) of the two samples:

$$D = \sup_x |F_n(x) - F_m(x)| \quad (1)$$

  Where $F_n(x)$ and $F_m(x)$ are the empirical CDFs of the reference and current samples, respectively.
- The Page-Hinkley Test is a sequential hypothesis testing method that can detect both gradual and sudden changes in the data distribution by monitoring the cumulative sum of deviations from a reference value. It detects shifts in the mean of a data stream:

$$PH_t = \sum_{i=1}^{t} (x_i - \mu - \delta) \quad (2)$$

  where x_i is the data point at time i, μ is the mean, and δ is the tolerance parameter.
  This approach can be effective for detecting both short-term and long-term drift, but may have a higher false positive rate than other methods.
- EWMA (Exponentially Weighted Moving Average): a statistical method that monitors changes in the mean or standard deviation of the incoming data stream and detects significant deviations:

$$S_t = \alpha X_t + (1 - \alpha) S_{t-1} \quad (3)$$

  Where S_t is the EWMA statistic at time t, X_t is the current data point, and α is the smoothing parameter.

- ADWIN (Adaptive Windowing): a non-parametric method that uses a sliding window to detect changes in the underlying distribution of the data by monitoring the variance of the data within the window. If a significant difference is found, a drift is signaled, and the window is reset.

- CUSUM (Cumulative Sum) a statistical control chart method that detects changes in the mean of the data by monitoring the cumulative sum of deviations from a reference value:

$$C_t = \max(0, C_{t-1} + X_t - k) \quad (4)$$

  Where C_t is the CUSUM statistic at time t, X_t is the data point, and k is the reference value.
- HDDM (Hoeffding's Drift Detection Method): a hypothesis testing method that detects changes in the distribution of the data by comparing the mean of two sliding windows of the data. Utilizes Hoeffding's inequality to determine if two sliding windows of data points have different distributions.
- DDM (Drift Detection Method) can detect drift in real-time with low computational overhead. DDM works by monitoring the error rate of a classification model over time and comparing it

to a threshold value. When the error rate exceeds the threshold, it indicates a possible drift, and the model can be updated to adapt to the new data distribution. The error rate ϵ and standard deviation σ are tracked, and a drift is signaled if:

$$\epsilon + \sigma > \epsilon_{min} + 2\sigma_{min} \quad (5)$$

These techniques are designed to detect different types of concept drift and have varying degrees of sensitivity and computational complexity. It's important to choose the appropriate technique based on the specific application and data characteristics.

Technique	Description	Pros	Cons
KS Test	Compares incoming data distribution to a reference, effective for sudden changes.	- Effective for sudden changes. - Provides statistical significance.	- May be less effective for gradual changes. - Higher false positives for subtle changes.
Page-Hinkley Test	Sequential hypothesis testing detecting both gradual and sudden changes.	- Effective for short-term and long-term drift. - Adapts to various drift types.	- May have a higher false positive rate.
EWMA	Monitors mean or standard deviation changes in incoming data stream.	- Detects significant deviations. - Provides adaptability to changes.	- May not handle complex drift patterns effectively.
ADWIN	Non-parametric method using a sliding window to detect changes in data distribution.	- Adapts to underlying distribution changes. - Real-time drift detection.	- Computational resources may be demanding.
CUSUM	Statistical control chart method detecting changes in the mean of the data.	- Effective for detecting mean changes. - Suitable for real-time detection.	- May require careful parameter tuning. - Sensitive to initial conditions.
HDDM	Hypothesis testing method detecting changes in the distribution of the data.	- Detects changes effectively. - Useful for specific contexts.	- Computational overhead may be a concern. - Sensitivity to parameter choices.
DDM	Window-related technique monitoring the error rate of a classification model over time.	- Simplicity and efficiency. - Real-time detection capabilities. - Robust to noise.	- May not perform optimally in all scenarios. - Limited by window size.

Table 2: Concept Drift Detection Techniques

DDM (Drift Detection Method) is a window-related technique used to detect concept drift in data streams. It works by maintaining a sliding window of the most recent data points and monitoring the changes in the data distribution over time. If the distribution changes significantly within the window, it signals the presence of a concept drift. Here are some reasons why DDM is considered in this contribution [17]:

- **Simplicity and Efficiency**: DDM is a simple and efficient method that can detect drift in real-time with low computational overhead. It works by monitoring the error rate of a classification

model over time and comparing it to a threshold value. When the error rate exceeds the threshold, it indicates a possible drift, and the model can be updated to adapt to the new data distribution. This simplicity and efficiency make DDM a practical and scalable approach for real-world applications.

- **Real-Time Detection**: DDM can detect drift in real-time, which is critical for intrusion detection and other time-sensitive applications. It can quickly identify changes in the data distribution and update the classification model to adapt to the new distribution. This real-time detection capability can improve the accuracy and timeliness of intrusion detection systems, which can help prevent or mitigate cyberattacks.
- **Adaptive Thresholding**: DDM uses an adaptive thresholding approach to adjust the detection sensitivity based on the number of observations and the significance level. This approach can reduce the false positive rate and improve the accuracy of drift detection, which is important for reducing the workload of security analysts and avoiding unnecessary alarms.
- **Robustness to Noise**: DDM is robust to noise and outliers in the data, which can be common in real-world intrusion detection applications. It can filter out noise and focus on significant changes in the data distribution, which can improve the reliability and effectiveness of the detection system.

DDM is a simple, efficient, and effective concept drift detection technique that can provide real-time detection and adaptive thresholding capabilities for intrusion detection and other real-time data analysis applications. Its robustness to noise and scalability make it a practical and reliable approach for detecting concept drift in a variety of settings. In our contribution, we used DDM for concept drift detection.

3.2 Online Incremental learning

After drift detection, a suitable drift adaptation algorithm needs to be implemented to handle the detected drifts and maintain high learning performance. Drift adaptation methods can be broadly classified into two main categories: incremental learning methods and ensemble methods [15]. Incremental learning methods update the model parameters gradually over time to adapt to concept drift, while ensemble methods combine multiple models to improve performance and handle concept drift [16].

Online incremental learning techniques are used to train models on continuously arriving data, where the model needs to be updated with each new sample. Some of the most popular online incremental learning techniques include [18]:

- *Online Sequential Extreme Learning Machine (OS-ELM):* OS-ELM is a popular online learning algorithm for training feedforward neural networks. It updates the network parameters using only one sample at a time and has a faster training speed compared to other neural network models. OS-ELM can handle large datasets and noisy data. It involves randomly assigning input weights and biases, then solving the output weights using a least-squares solution:

$$\mathbf{H}\beta = \mathbf{T} \tag{6}$$

  Where H is the hidden layer output matrix, β is the output weight matrix, and T is the target matrix.
- *Incremental and Decremental Support Vector Machines (ICU-SVM):* ICU-SVM is an online learning algorithm for classification and regression tasks. It updates the SVM model parameters using only one sample at a time and can handle non-stationary data. ICU-SVM has a high accuracy and is computationally efficient.
- *Hoeffding Tree*: Hoeffding Tree is an online learning algorithm for classification and regression tasks. It builds a decision tree incrementally using a statistical test to determine

when to split the nodes. Hoeffding Tree can handle large datasets and noisy data and has a fast processing speed. It uses the Hoeffding bound to determine the minimum number of samples required to choose the best splitting attribute with high probability:

$$G(X_i) - G(X_j) > \epsilon \quad (7)$$

Where G is the split evaluation measure, and ϵ\epsilonϵ is the Hoeffding bound.

- *Stochastic Gradient Descent (SGD):* SGD is a popular online learning algorithm used for a variety of machine learning tasks. It updates the model parameters based on the gradient of the loss function with respect to the parameters. SGD is computationally efficient and can handle large datasets:

$$\theta_{t+1} = \theta_t - \eta \nabla L(\theta_t; x_t, y_t) \quad (8)$$

Where θ represents model parameters, η is the learning rate, and L is the loss function.

- *Adaptive Learning Rate Methods*: Adaptive learning rate methods are a class of online learning algorithms that adjust the learning rate based on the characteristics of the data. These methods include Adagrad, Adadelta, RMSProp, and Adam. Adaptive learning rate methods can improve the convergence rate and accuracy of the model.

Technique	Description	Pros	Cons
OS-ELM	Efficient online learning algorithm for training feedforward neural networks.	- Fast training speed. - Suitable for large datasets and noisy data.	- Limited interpretability of neural networks. - May not perform well with highly imbalanced datasets.
ICU-SVM	Online learning algorithm for classification and regression tasks.	- High accuracy. - Computationally efficient. - Handles non-stationary data.	- May require tuning of hyperparameters. - Limited adaptability to complex data distributions.
Hoeffding Tree	Online learning algorithm for classification and regression tasks, building decision trees incrementally.	- Handles large datasets and noisy data. - Fast processing speed.	- Limited interpretability of decision trees. - May struggle with highly imbalanced datasets.
SGD	Popular online learning algorithm used for various machine learning tasks.	- Computationally efficient. - Suitable for large datasets.	- May require careful tuning of hyperparameters. - Sensitive to feature scaling. - Convergence may be affected by noisy data.
Adaptive Learning Rate Methods	Class of online learning algorithms adjusting learning rate based on data characteristics.	- Improves convergence rate and accuracy of the model. - Handles changing data characteristics effectively.	- May require tuning of adaptive learning rate methods. - Sensitive to hyperparameter choices. - Performance may vary across different datasets.
Online Passive-Aggressive Algorithm	Online learning algorithm for classification tasks, updating model parameters based on sample margin.	- Computationally efficient. - Suitable for high-dimensional data.	- May require tuning of hyperparameters. - Limited interpretability. - Sensitivity to feature scaling. - May be affected by noisy data.
Bayesian Methods	Class of online learning algorithms updating model parameters based on Bayesian inference.	- Handles uncertainty in data. - Improves model generalization.	- May require careful tuning of Bayesian methods. - Computational overhead may be a concern. - Limited interpretability for some Bayesian models.
Online Random	Extension of traditional random	- Handles high-dimensional	- May require careful tuning of

	capable of handling concept drift.	- Detects and adapts to concept drift. - Low computational cost. - Suitable for large-scale datasets.	- Limited interpretability of the ensemble. - Sensitive to noisy data and outliers. - Resource-intensive during training.

Table 3 : Online Incremental Learning Techniques

- *Online Passive-Aggressive (PA) Algorithm*: The PA algorithm is a popular online learning algorithm for classification tasks. It updates the model parameters based on the margin of the sample and the prediction of the current model. The PA algorithm is computationally efficient and can handle high-dimensional data:

$$\theta_{t+1} = \theta_t + \tau y_t x_t \quad (9)$$

Where τ is the step size determined by the loss function, and y_t and x_t are the label and feature vector of the current sample.

- *Bayesian Methods*: Bayesian methods are a class of online learning algorithms that update the model parameters based on the Bayesian inference framework. These methods include online Bayesian linear regression, online Bayesian logistic regression, and online Bayesian neural networks. Bayesian methods can handle uncertainty in the data and improve the generalization ability of the model.
- *Online Random Forests*: Online random forests are an extension of the traditional random forest algorithm for online learning. They update the model by adding new decision trees to the forest based on the arriving data. Online random forests can handle concept drift and noisy data:

$$\hat{y} = \frac{1}{n} \sum_{i=1}^{n} T_i(x) \quad (10)$$

Where ^y is the predicted output, n is the number of trees, and T_i is the i-th decision tree.

Online random forests [12] are a powerful and effective technique for incremental learning. One of the main advantages of online random forests is their ability to handle high-dimensional and noisy data in an online and dynamic environment. They can handle both categorical and continuous data and have the ability to detect and adapt to concept drift, which is a common problem in online learning [19]. Additionally, online random forests have a low computational cost and can be trained on large-scale datasets. Overall, online random forests are a flexible and robust technique that can handle a wide range of data types and learning scenarios, making them one of the best choices among the listed techniques [20]. In our contribution we opted for Online random forests for online incremental learning.

4. PROPOSED APPROACH

We propose an intrusion detection approach, named DDM-ORF, that combines concept drift detection based on Drift Detection Method and online incremental learning based on Online Random Forest to improve the accuracy and efficiency of intrusion detection systems.

The Figure 1 illustrates the operational procedure utilized in the presented approach. The initial step involves data preprocessing, which converts the raw streaming data into a suitable format for subsequent processing. The next stage employs a drift detection technique known as DDM to identify the presence of concept drift. Finally, the ORF model is implemented to determine the class label of the streaming data.

The proposed approach consists of five main components: (1) Data Collection, (2) Data preprocessing, (3) Random Forest based Model Training, (4) DDM based drift detection, and (5) Online Random Forest based Incremental learning that updates the model in real-time.

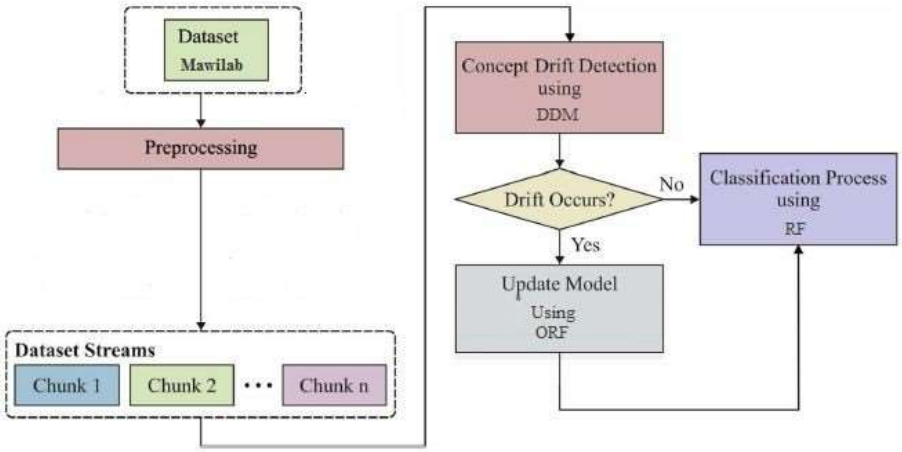


Figure 1: Proposed Approach

4.1 Data Collection

The Mawilab dataset [14] is a publicly available dataset developed by the Fukuda Laboratory at the University of Tokyo in Japan. The Mawilab dataset is a collection of network traffic data that has been collected over several years from various sources, including honeypots, darknets, and internet service providers. The dataset contains both benign and malicious network traffic data, making it a valuable resource for researchers studying network security and cyber-attacks. The dataset was initiated in 2004, and since then, it has been continuously updated with new network traffic data and improved analysis techniques.

In our contribution, as the Mawilab dataset is collected from web traffic data, we use a web scrapper (BeautifulSoup) to extract specific data from the HTML files that represent the web traffic. This data is then processed, transformed, and stored in Azure in streaming.

Storing the Mawilab dataset on Azure in streaming involves using Azure Stream Analytics, a cloud-based service for real-time data processing. This approach enables the dataset to be stored in real-time, as the network traffic data is received from Mawilab.

To store the dataset in streaming, we create an Azure Stream Analytics job in the Azure portal. This job acts as a pipeline that receives the network traffic data from Mawilab in real-time and stores it in Azure. The input for the Azure Stream Analytics job is configured as an Azure Event Hub, which is a scalable and real-time data streaming platform. The output is configured to store the data in various Azure storage solutions: Azure Blob Storage or Azure Data Lake Storage. Then, we define the queries for the Azure Stream Analytics job. The queries specify how the data received from Mawilab will be processed and transformed before being stored in Azure.

Once the Azure Stream Analytics job is configured, it begins to receive the network traffic data from Mawilab in real-time and stores it in Azure in streaming. By using this approach, we gain insights from the data in real-time and respond to potential security threats more quickly.

4.2 Data Preprocessing

This step is about Mawilab data preprocessing using Spark on Azure HDInsight Spark cluster [21]. This step consists in [23]:

- Data ingestion: The Mawilab dataset is ingested into the Spark cluster using Azure Blob Storage method. Then, data is converted from CSV into Apache Parquet format.
- Data cleaning: The Mawilab dataset contains missing and invalid data, which needs to be cleaned before analysis. We use various functions Spark provides for data cleaning: drop() and fillna(). This step also involves eliminating redundancies in the data with dropDuplicates() function.
- Data transformation: Once the data is cleaned, it undergoes several transformation steps to prepare it for machine learning algorithms. This includes feature engineering, feature selection, and encoding categorical variables. We leverage Apache Spark's functionalities to efficiently handle large-scale data transformation.

  ❖ Feature Engineering

  Feature engineering involves creating new features from the existing ones to enhance the predictive power of the machine learning model. This step includes several sub-tasks:

  VectorAssembler: Combines multiple columns into a single feature vector. This is useful for algorithms that expect a single vector input. The `VectorAssembler` function in Spark is used as follows:

```
from pyspark.ml.feature import VectorAssembler
# Define the input columns to be assembled into a feature vector
input_columns = ['feature1', 'feature2', 'feature3']
assembler = VectorAssembler(inputCols=input_columns, outputCol='features')
# Transform the dataset
transformed_data = assembler.transform(cleaned_data)
```

  This transformation combines specified input columns into a single feature vector column named 'features'.

  StringIndexer: Converts categorical variables into numerical indices. This is essential for machine learning algorithms that cannot handle categorical data directly. The `StringIndexer` function in Spark is used as follows:

```
from pyspark.ml.feature import StringIndexer
# Define the column to be indexed
indexer = StringIndexer(inputCol='category_column',
outputCol='category_index')
# Fit the indexer to the data and transform it
indexed_data = indexer.fit(cleaned_data).transform(cleaned_data)
```

  This transformation converts the 'category_column' into numerical indices, stored in 'category_index'.

  Polynomial Expansion: Generates polynomial features from the existing features, increasing the feature space to capture non-linear relationships. In Spark, this can be done using the `PolynomialExpansion` function:

```
from pyspark.ml.feature import PolynomialExpansion
# Define the polynomial expansion degree
poly_expansion = PolynomialExpansion(inputCol='features',
outputCol='poly_features', degree=2)
# Transform the dataset
expanded_data = poly_expansion.transform(transformed_data)
```

  This transformation creates polynomial features of degree 2 from the input feature vector.

Normalization: Scales the feature vectors to have unit norm. This ensures that all features contribute equally to the distance calculations in algorithms like k-NN or SVM. The `Normalizer` function in Spark is used as follows:

```
from pyspark.ml.feature import Normalizer
# Define the normalizer
normalizer = Normalizer(inputCol='features', outputCol='norm_features',
p=2.0)
# Transform the dataset
normalized_data = normalizer.transform(expanded_data)
```

This transformation normalizes the feature vectors to have unit L^2 norm.

- ❖ Feature Selection

Feature selection is the process of identifying the most relevant features for the machine learning model. This step helps in reducing the dimensionality of the data, improving model performance, and avoiding overfitting. Several techniques can be used for feature selection:

Chi-Square Test: Evaluates the independence between each feature and the target variable, selecting features that are statistically significant. In Spark, the `ChiSqSelector` function is used as follows:

```
from pyspark.ml.feature import ChiSqSelector
# Define the Chi-Square selector
selector = ChiSqSelector(numTopFeatures=10, featuresCol='features',
outputCol='selected_features', labelCol='label')
# Fit the selector to the data and transform it
selected_data = selector.fit(normalized_data).transform(normalized_data)
```

This selects the top 10 features that have the highest Chi-Square statistic with the target variable.

Correlation Matrix: Computes the correlation between features and the target variable, selecting features with high absolute correlation values. This can be done using Spark's `Correlation` function:

```
from pyspark.ml.stat import Correlation
# Compute the correlation matrix
correlation_matrix = Correlation.corr(normalized_data,
'features').head()[0]
# Select features based on a correlation threshold
selected_features = [i for i in range(len(correlation_matrix)) if
abs(correlation_matrix[i]) > 0.1]
```

This selects features with an absolute correlation value greater than 0.1.

Feature Importance from Tree-Based Models: Uses the feature importance scores from tree-based models like Random Forests to select the most important features. This can be achieved using Spark's `RandomForestClassifier`:

```
from pyspark.ml.classification import RandomForestClassifier
# Train a Random Forest model
rf = RandomForestClassifier(featuresCol='features', labelCol='label')
model = rf.fit(normalized_data)
# Extract feature importance scores
feature_importances = model.featureImportances
# Select features based on importance scores
selected_features = [i for i, importance in enumerate(feature_importances)
if importance > 0.01]
```

This selects features with importance scores greater than 0.01.

By incorporating these steps into the data transformation process, we ensure that the data is adequately prepared for machine learning algorithms, enhancing the predictive power and performance of the models.

- Data storage: Once the data is preprocessed and transformed, it is stored in Parquet format in Azure Blob Storage for further analysis.

Mawilab data preprocessing using Spark on Azure HDInsight Spark cluster involves leveraging Spark's data processing and transformation capabilities, along with using Azure Blob Storage service for data storage and ingestion. Additionally, using Parquet format for data storage helps improve query performance and reduce storage costs.

4.3 Random Forest based Model Training

RF works by building an ensemble of decision trees using a bagging technique, where each tree is trained on a randomly sampled subset of the data and a randomly sampled subset of the features. Given a training set of N data points, the RF algorithm builds T decision trees, each of which predicts the class label of a new data point based on a majority vote of its leaf nodes [32]:

$$f(x)=argmax_c \sum_{t=1}^{T} I(ht(x) = c) \quad (11)$$

where f(x) is the predicted class label for a new data point x, h_t(x) is the class label predicted by the t^{th} decision tree, and \mathbb{I}(\cdot) is the indicator function that returns 1 if its argument is true and 0 otherwise.

The algorithm [33] works by creating a forest of decision trees, where each tree is built on a random subset of the original data. This process is called bagging or bootstrap aggregating, and it helps to reduce the variance of the model and prevent overfitting.

For classification tasks, the algorithm randomly selects a subset of features at each split point, instead of using all the available features, which further helps to reduce the correlation between the trees and improve the accuracy of the model. During the prediction phase, the algorithm takes a new data point and passes it through each decision tree in the forest, and each tree produces a classification or regression output.

The final prediction is then made by aggregating the outputs of all the trees, either by majority voting in the case of classification or by taking the average in the case of regression.

The tree is created using a top-down approach [27] (see Figure 3), where the algorithm starts with the entire dataset and repeatedly splits it into smaller subsets based on the value of a single feature that maximizes the information gain or minimizes the impurity. Information gain measures how much a particular feature contributes to reducing the uncertainty in the classification or regression task, while impurity measures the degree of disorder or randomness in the data.

In our contribution, a trained machine learning pipeline using Random Forest algorithm is loaded from Azure Blob Storage. Live data is classified using the pipeline and predictions are then saved in Azure Data Lake.

Algorithm: Random Forest

```
#Inputs
T: number of trees
d_max: maximum tree depth
m: number of features
Training dataset: a set of (x, y) pairs
#Initialize an empty forest F with T trees
F ← {}
for i = 1 to T do
t_i ← create_new_tree(d_max, m)
F ← F ∪ t_i
#For each incoming training instance
for each tree t_i in F do
S_i ← sample_feature_subset(m)
n_i ← find_leaf_node(t_i, x)
update_leaf_statistics(n_i, y)
if depth(n_i) < d_max and enough_instances_to_split(n_i) then
(n_i_l, n_i_r) ← split_node(n_i, S_i)
add_child_nodes(n_i, n_i_l, n_i_r)
#To make a prediction
for each tree t_i in F do
n_i ← find_leaf_node(t_i, x_new)
compute_leaf_prediction(n_i)
```

Figure 2: Random Forest Algorithm

Algorithm: Decision Tree

```
#Define the tree node
S the set of samples
X the set of features
Y the set of labels
#Calculate the Gini impurity
Gini(S) = 1 - Σ p_i^2
where p_i is the proportion of samples in S with label i
For each feature x in X
#Calculate the Gini impurity of splitting S on x
Let S_l be the set of samples in S where x <= t
where t is a threshold value for feature x
Let S_r be the set of samples in S where x > t
Gini_l = Gini(S_l)
Gini_r = Gini(S_r)
Gini_split = (|S_l|/|S|) * Gini_l + (|S_r|/|S|) * Gini_r
#Calculate the information gain of splitting S on x
Info_gain = Gini(S) - Gini_split
#Keep track of the feature x with the highest gain
If the gain is less than a threshold value
Stop splitting and create a leaf node with label y
Otherwise,
Create an internal node with feature x and threshold value t
```

Figure 3: Decision Tree Algorithm

4.4 DDM based Drift Detection

Let y be the target variable we want to predict, and let f(x; θ) be the predictive model that maps the input variable x to the predicted target value y, where θ is the set of model parameters [34]. At each time step t, we observe a new data point (xt, yt) and compute the prediction error εt as:

$$\varepsilon t = yt - f(xt;\ \theta) \quad (12)$$

We then update the mean error rate m and the standard deviation of the error rate s as follows:

$$m(t) = m(t\text{-}1) + (\varepsilon t - m(t\text{-}1))/(t+1) \quad (13)$$

$$s(t) = sqrt(s\text{^}2(t\text{-}1) + (\varepsilon t - m(t\text{-}1))*(\varepsilon t - m(t))) \quad (14)$$

The drift measure d(t) at time t is defined as:

$$d(t) = |\varepsilon t - m(t)|/s(t) \quad (15)$$

If d(t) exceeds a pre-defined threshold ω, we declare a drift at time t.
The threshold ω can be computed based on the desired false positive rate (α) and false negative rate (β) as:

$$\omega = sqrt((1/2) * log(2/\alpha) * (1/\beta)) \quad (16)$$

The idea behind this threshold is to control the risk of false positives and false negatives while detecting changes in the data stream.

The different steps of the DDM based Drift Detection algorithm are as follows:

1. Initialization:
   - The algorithm initializes various parameters such as time t, evidence x, threshold a, drift rate v, non-decision time T, standard deviation of noise σ, and window size w.
2. Processing Loop:
   - The loop runs until a decision is made.
3. Increment Time:
   - Time t is incremented by a small time step dt.
4. Evidence Accumulation:
   - The change in evidence dx is calculated using the drift rate, time step, and a random value dW sampled from a normal distribution.
5. Update Evidence:
   - The evidence x is updated by adding dx.
6. Moving Average and Standard Deviation:
   - Evidence values are stored in a moving window. The moving average and standard deviation of recent evidence values are calculated to smooth out noise and maintain stability.
7. Adjust Threshold:
   - The decision threshold a is dynamically adjusted based on the moving standard deviation of recent evidence values.
8. Threshold Check:
   - The algorithm checks if the absolute value of x exceeds the threshold a. If so, a decision is made based on the sign of x.
9. Non-Decision Time:
   - Non-decision time T is added to the total time t.
10. Output:
    - The algorithm outputs the decision and response time.

Algorithm : Drift Detection Method

```
# Initialize variables
t = 0                    # Start time
x = 0                    # Starting evidence
a = initial_threshold    # Starting threshold
v = initial_drift_rate   # Starting drift rate
T = initial_non_decision_time  # Starting non-decision time
sigma = initial_sigma    # Starting standard deviation of the noise
w = initial_window_size  # Window size for moving average
evidence_window = []     # List to store recent evidence values
# Repeat until decision is made
while not decision_made:
    # a. Increment time
    t += dt
    # b. Calculate evidence accumulation
    dW = random.normalvariate(0, 1)  # dW is a random value from N(0, 1)
    dx = v * dt + sigma * dW         # dx is the change in evidence over time dt
    # c. Update evidence
    x += dx
    # Store evidence in the moving window
    evidence_window.append(x)
    if len(evidence_window) > w:
        evidence_window.pop(0)
    # Calculate moving average and standard deviation of recent evidence values
    if len(evidence_window) > 1:
        moving_avg = sum(evidence_window) / len(evidence_window)
        moving_std = (sum([(xi - moving_avg) ** 2 for xi in evidence_window]) / (len(evidence_window) - 1))
** 0.5
    # Adjust decision threshold based on recent evidence
    a = initial_threshold + k * moving_std  # k is a scaling factor for threshold adjustment
    # d. Check if decision threshold is reached
    if abs(x) >= a:
        decision_made = True
        decision = 'positive' if x > 0 else 'negative'
    else:
        continue
    # e. Add non-decision time
    t += T
# Output decision and response time
output_decision = decision
response_time = t
return output_decision, response_time
```

Figure 4: DDM pseudo Algorithm

Figure 5 shows the flow chart of our DDM based Drift Detection contribution. Mawilab data arrives over time in batches. D_t represents de t^{th} batch. Each batch contains a number of instances.

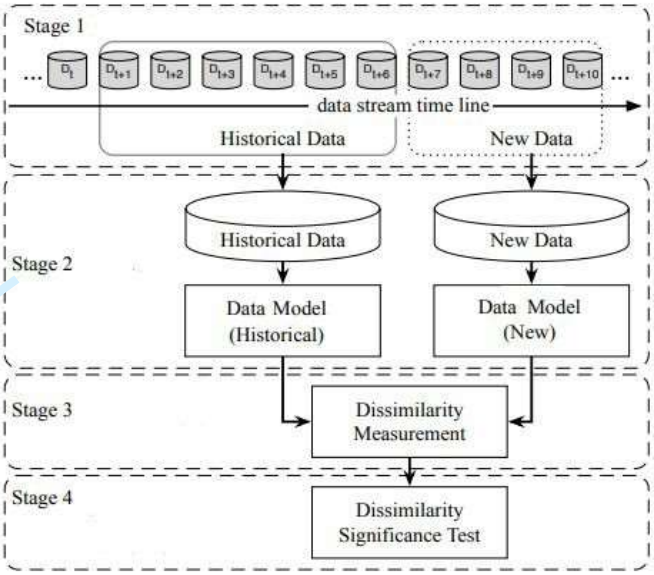


Figure 5: DDM based Drift Detection

In this algorithm [35], the first stage involves a time window that is used to monitor the overall error rate of the system. Whenever a new data instance becomes available, the algorithm checks if there has been a significant increase in the error rate within the time window. If the observed error rate change exceeds a certain confidence level, the algorithm starts building a new learner while still using the old one for predictions. If the change exceeds a certain drift level, the old learner is replaced with the new one for all future prediction tasks. To determine the error rate, the algorithm relies on a classifier to make predictions, which is considered the second stage of the algorithm. The online error rate is then used in the third stage to calculate test statistics. Finally, in the fourth stage, a hypothesis test is conducted by estimating the distribution of the online error rate and calculating warning and drift thresholds.

4.5 Online Random Forest Incremental Learning

Online Random Forest is an extension of Random Forest that allows the model to learn incrementally, i.e., it can update the model parameters with new data points without the need to retrain the entire model from scratch**.** Online Random Forest (ORF) is a type of incremental learning algorithm that can be used to handle concept drift in streaming data. ORF is based on the popular Random Forest (RF) algorithm, but instead of building a static forest on a fixed dataset, ORF incrementally updates the forest as new data arrives [20].

The ORF algorithm extends the RF algorithm to handle streaming data by incrementally updating the ensemble of decision trees as new data arrives. Specifically, ORF maintains a sliding window of size W that contains the most recent W data points and updates the ensemble after each new data point x_t arrives. The ORF algorithm consists of the following steps [25]:

1. Add x_t to the sliding window and remove the oldest data point if the window size exceeds W.
2. For each decision tree t in the ensemble, update the tree using the following procedure:
   - Choose a random subset of features for the tree (this is called the "random subspace" method).
   - Use the incremental learning algorithm to update the tree with x_t.

3. If the number of trees in the ensemble is less than T, add a new decision tree to the ensemble initialized with x_t.

The prediction of ORF on a new data point x is obtained by taking a majority vote of the predictions of each decision tree in the ensemble:

$$f(x)=argmax_c \sum_{t=1}^{T} I(ht(x) = c) \quad (17)$$

Where h_t(x) is the class label predicted by the t^{th} decision tree for the new data point x. ORF uses an incremental learning algorithm to update each decision tree in the ensemble, which allows the algorithm to update the model efficiently as new data arrives. Additionally, ORF uses a random subspace method to reduce the correlation between decision trees, which helps to improve the diversity of the ensemble and avoid overfitting.

The algorithm [28] starts by training an initial random forest model on a portion of the available data. As new data becomes available, it feeds it into the model one observation at a time. For each new observation, it determines which leaf node it falls into in each tree of the random forest. The algorithm updates the statistics associated with each leaf node for each tree. These statistics include the count of observations in the node, the sum of the response variable, and the sum of the squared response variable.

The algorithm recalculates the prediction for each tree using the updated statistics for each leaf node, combines the predictions from each tree to obtain the final prediction for the new observation. The algorithm periodically, re-evaluates the performance of the model on a hold-out set of data and consider pruning the trees that are not contributing significantly to the model's accuracy.

Algorithm : Online Random Forest

Initialization:

- **F**: A set of decision trees.
- **Procedure:** For each incoming data point x_i:
- For each tree T_j in F:
  - Let l be the leaf node in T_j that x_i belongs to.
  - If the number of data points in ll is less than a pre-defined threshold nnn:
    - Add x_i to l.
- If the number of data points in l exceeds the threshold n:
  - Randomly select a subset SSS of the data points in l.
  - Grow a new tree T_k using a decision tree algorithm on S.

Add T_k to F.

Figure 6: Online Random Forest pseudo algorithm

The algorithm Online Random Forest (Figure 6) starts by initializing a set of decision trees F.

Processing Incoming Data: For each incoming data point x_i, the algorithm processes it through each tree T_j in the forest F.

Finding the Leaf Node: For each tree T_j, the algorithm finds the leaf node l where the data point x_i belongs. This is typically done by traversing the tree from the root to a leaf node based on the features of x_i.

Updating the Leaf Node: If the number of data points in the leaf node l is less than a pre-defined threshold n, the data point x_i is simply added to l.
If the number of data points in l exceeds the threshold n, the algorithm proceeds to handle the overflow.
Handling Overflow: A subset S of the data points in l is randomly selected. This subset is used to grow a new decision tree T_k.
The new tree T_k is grown using a standard decision tree algorithm applied to the subset S.
The new tree T_k is then added to the forest F, expanding the ensemble.

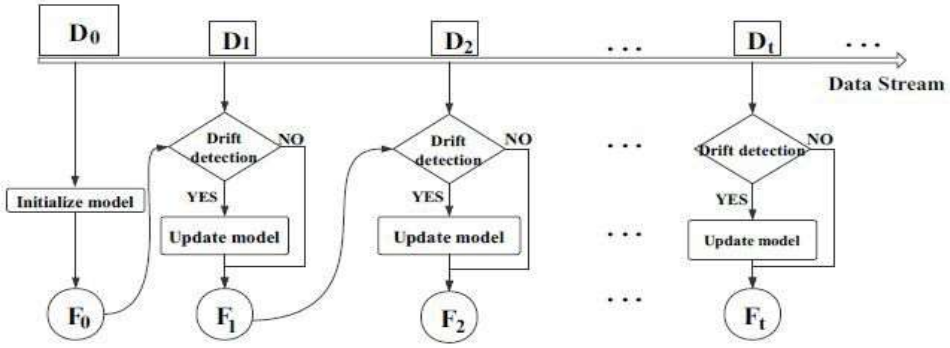


Figure 7: Incremental learning flow chart

Figure 7 shows the flow chart of our online incremental learning contribution. The process initiates with the continuous monitoring of the data stream for signs of concept drift using the Drift Detection Method (DDM). This method operates by tracking the error rate of the classifier over time and comparing it to predefined threshold values. When the error rate exceeds these thresholds, it indicates a potential drift in the underlying data distribution.

Upon detecting a concept drift, the system triggers the online incremental learning mechanism to adapt to the new data patterns. The classifier update is executed using the Online Random Forest (ORF) algorithm, which is particularly well-suited for real-time data processing due to its ability to incrementally update the model without needing to retrain from scratch.

The ORF algorithm operates by adjusting the ensemble of decision trees in the random forest. When new data arrives, each tree in the forest is updated based on the new instances. Specifically, the algorithm employs techniques such as incremental learning of decision trees, where nodes are split or pruned based on the new data, and the weights of the trees are adjusted to reflect the recent changes in the data distribution.

Furthermore, the ORF algorithm integrates mechanisms to handle various types of concept drift, including abrupt, gradual, and recurrent drifts. It does this by maintaining a sliding window of recent instances and periodically evaluating the performance of each tree within this window. Trees that consistently perform poorly are replaced or updated more aggressively, ensuring that the forest remains robust and adaptive to the evolving data landscape.
This approach ensures that the classifier remains accurate and efficient in the face of continuous data stream changes, providing timely and reliable intrusion detection. The integration with Apache Spark Structured Streaming further enhances the system's capability to process large-scale data in a distributed manner, making it suitable for real-world applications with high throughput and low latency requirements.

5. EXPERIMENTATION AND DISCUSSION

To evaluate the detection capabilities of the proposed DDM-ORF model, a series of experiments were conducted using the Mawilab dataset, and a variety of evaluation metrics: precision, recall, accuracy, and F-measure.

The MAWILab dataset is comprised of labeled traffic flows indicating whether they are anomalous or not. This dataset employs four different anomaly detection methods (Hough transform, Gamma distribution, Kullback-Leibler divergence, and Principal Component Analysis 'PCA') [36]. The traffic flows in the dataset are classified into four main categories as follows :

- Anomalous: Traffic flows that are deemed abnormal and are detected by the employed anomaly detectors.
- Suspicious: Traffic flows that are likely to be anomalous but are not clearly detected by the anomaly detectors.
- Notice: Traffic flows that are normal but have been reported by one or more of the anomaly detectors.
- Benign: Normal traffic flows that have not been reported or detected by any of the anomaly detectors.

In their work, [30] introduced a taxonomy that outlines the characteristics of backbone traffic anomalies. The MAWILab dataset leverages this taxonomy by including a dedicated field to provide further insights into the nature of anomalies. The table presented below illustrates the various taxonomies employed in the dataset.

Taxonomy	Description
"Unk", "empty"	Unknown labels
"ttl_error","hostout","netout", "icmp_error"	Other labels
"alphflHTTP","ptmpHTTP","mptpHTTP","ptmplaHTTP","mptplaHTTP"	HTTP
"ptmp","mptp", "mptmp"	Multi Points
"alphfl","malphfl","salphfl","point_to_point", "heavy_hitter"	Alphaflow
"ipv4gretun", "ipv46tun"	IPv6 tunneling
"posca", "ptpposca"	Port scan
"ntscIC", "dntscIC"	Network scan ICMP
"ntscUDP", "ptpposcaUDP"	Network scan UDP
"ntscACK","ntscSYN","sntscSYN","ntscTCP","ntscnull","ntscXmas","ntscFIN", "dntscSYN"	Network scan TCP
"DoS","distributed_dos","ptpDoS","sptpDoS","DDoS", "rflat"	DoS

Table 4: Description of Taxonomies

We conduct Data preprocessing using a Microsoft HDInsight cluster running Apache Spark Structured Streaming. Structured Streaming is built on top of Spark SQL engine which gives us exactly once delivery and providing end-to-end reliability.

When deploying an HDInsight cluster, Azure uses the Hortonworks Data Platform (HDP) which is powered by Apache Hadoop. HDP is a massively scalable and open source, it is used for storing, processing and analyzing big data. It is designed to handle multiple data sources and formats with a user-friendly dashboard. HDP consists of a set of Hadoop projects including Storm, Spark, Ambari, etc. Once the cluster is deployed, HDInsight provides multiple options for the user to choose from. In addition, Azure provides multiple visualization tabs to track and

monitor the cluster performance for processing, storage and bandwidth.

When we transform CSV files into Parquet format, we observe benefits in both cost and performance. By using Parquet, we not only reduce the time spent waiting for data to be scanned and processed, but also lower storage expenses [33]. The MAWILab dataset is updated frequently, with new files being added to their website on a regular basis. Our web scraper promptly ingests any new files that become available. To estimate the storage gain for each new file, we use an average file size since the file sizes can vary. Table 5 displays the size differences between the old and new formats after conversion to Apache Parquet.

Average size (CSV)	Average size (Parquet)	Speedup
9.5 KB	6.5 KB	X1.46

Table 5: Format Conversion

Structured Streaming is a stream processing model introduced in Apache Spark version 2.0. It is scalable and fault-tolerant, and it uses Data Frame API to simplify the development of real-time Big Data applications. The key concept of Structured Streaming is to treat an incoming stream of live data as a table that is being appended by new rows. For Structured Streaming, the idea is to append data to an unbounded input table. Users can specify how often these tables are appended by using triggers. If a trigger is set to one second, then Structured Streaming collects streaming data for one second and then append all of it to the input table. And this process reruns depending on a trigger interval set by the user. Afterwards, a query is run on the data and the result of that query is saved to the result table or output table. The result table is written every time to an output sink that is specified by the user and can be a database, storage space or another streaming job.

With Structured Streaming, it is possibleto select relevant columns from a dataset and ignore the rest of unselected columns. Aggregation and filter operations can be applied on data too. This includes filtering datasets based on one or many columns, also aggregation on data can be applied to extract only relevant fields. Spark's Structured Streaming API offers a solution to eliminate duplicate rows from a continuous stream of data by utilizing a unique identifier column. By keeping and storing data from previous records, Structured Streaming is able to filter out duplicate records.

We begin by reading Parquet files as a stream, and then selecting specific features that aid in the detection process. To reduce computational burden, unnecessary features such as anomaly_id and label (empty columns) are eliminated. Additionally, instead of deleting records that contain missing fields, we fill them with default values in order to create a complete dataset, and we eliminate any duplicate records.

After data preparation, we proceed to load the RF classifier, the DDM detector, and the ORF classifier, and transform incoming data to obtain predictions for each record. A Pipeline is a sequence of stages where each stage is either a Transformer or an Estimator, and these stages are executed in a specific order. In our work, we utilized two transformers: String Indexer and Vector Assembler. The String Indexer encodes a string column of labels into a column of label indices, which are ordered by label frequencies, with the most frequent label receiving index 0. The Vector Assembler is a transformer that combines a specified list of columns into a single vector column. This transformer is helpful in consolidating raw features and features produced by different feature transformers into a single feature vector. We then use a portion of the historic data obtained from the Mawilab dataset to train the pipeline model through the aforementioned stages.

The pipeline model is saved in Azure Blob Storage. The process of creating a pipeline model is

done offline. Once a concept drift is detected, the RF model is automatically updated using ORF incremental learning to train on new data, and the updated model is exported to replace the old model. Predictions are then written to a specified output location or sink, with the format of the output data (such as Parquet, JSON, etc.) being specified, along with a checkpoint location to ensure fault-tolerance.

To conduct our experiments, we utilize Microsoft Azure HDInsight, which runs on Linux virtual machines and Spark version 2.0 on top of YARN, using Jupyter Notebook and the Python API (Pyspark). During the streaming job, Spark ingests data, classifies it, detects concept drift, updates the model, and writes each file to the designated sink. The sink may be a file directory, a database, or even another Spark job.

5.1 Experiment 1

In this experiment, we evaluate the proposed DDM-ORF approach in terms of classification performance. Apache Spark offers a collection of metrics that can be used to assess the performance of machine learning models. In our work, we employed the following metrics to evaluate the performance of our pipeline:

- Accuracy: Measures precision across all labels.

$$AC = \frac{TP+TN}{TP+FN+TN+FP} \quad (18)$$

- Precision: Measures the proportion of correct classified labels over all labels.

$$P = \frac{TP}{TP+FP} \quad (19)$$

- Recall: Measures the proportion of correct labels correctly classified over all positive labels.

$$R = \frac{TP}{TP+FN} \quad (20)$$

- F-Measure: Measures the average of precision and recall.

$$FM = 2 * \frac{P*R}{P+R} \quad (21)$$

Where TP = True Positives, TN = True Negatives, FP = False Positives and FN = False Negatives.

The results generated by the pipeline can be seen in Table 6 below:

Accuracy	Precision	Recall	F-measure
99,96 %	99,93 %	99,95 %	99,94 %

Table 6: DDM-ORF Results

The Receiver Operating Characteristic curve (ROC curve) is a technique that can be employed to assess an individual's capabilities in distinguishing between groups. Derived from the ROC curve, the numerical measure used to estimate the curve is the Area Under the ROC Curve (AUC). This measure can be interpreted as the probability that the model makes correct predictions. The AUC serves as an effective means of summarizing the overall diagnostic accuracy of a test. Typically:

- If AUC = 0.5: the diagnostic result is normal.
- If 0.5 < AUC < 0.7: the result is not very significant.
- If 0.7 < AUC < 0.8: the result is considered acceptable.
- Results are deemed excellent if 0.8 < AUC < 0.9.

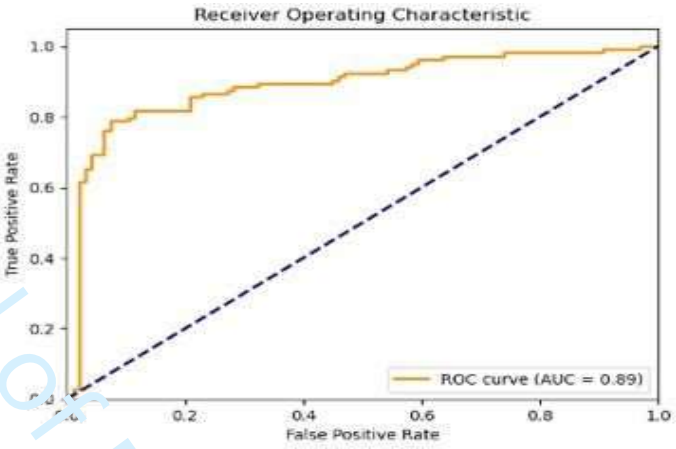


Figure 8 : DDM-ORF ROC Curve

The confusion matrix [36], can be regarded as a tool with the capability to analyze whether a classifier can successfully recognize tuples from different classes. True Positive and True Negative values provide insights into the true nature of the classifier when classifying data. On the other hand, False Positive and False Negative values provide information when the classifier is incorrect in classifying data.

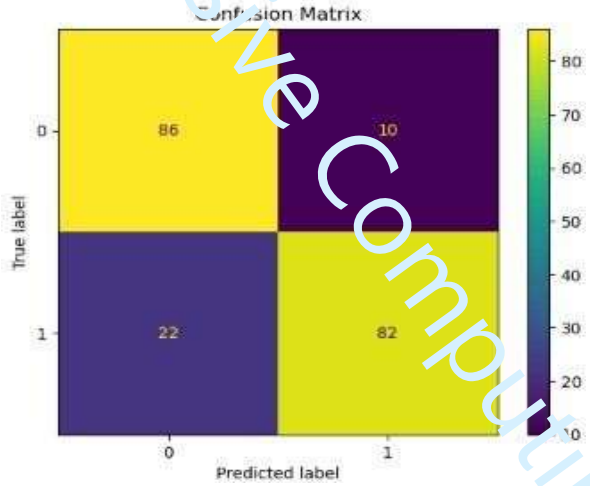


Figure 9: Confusion Matrix

To evaluate the effectiveness of the proposed DDM-ORF approach, we compare it against several state-of-the-art methods: Ensemble Incremental Learning [1], Adaptive Class Incremental Learning [2], and Active Incremental Learning [3]. These methods were chosen based on their relevance and reported performance in handling concept drift and online incremental learning. The section below also includes a comparison between our DDM-ORF proposed approach and the approach proposed in [12]. Authors in [12] proposed an intrusion detection system that uses deep learning techniques to analyze network traffic and detect anomalous and suspicious traffic. The authors used a deep neural network architecture called a convolutional neural network (CNN) to analyze features and classify network traffic. The CNN consisted of several layers, including convolutional layers, pooling layers, and fully connected layers. The authors also used dropout regularization to prevent overfitting. They trained the CNN on the training set and tuned the

hyperparameters using the validation set. They then evaluated the performance of the system on the testing set.

Table 7 and Figure 10 present a comprehensive comparison of the proposed DDM-ORF method against the state-of-the-art methods, including the CNN-based model proposed in [12]. Metrics such as accuracy, precision, recall, F1-score, and processing time were used to evaluate the performance.

Method	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**Processing Time**
Proposed DDM-ORF	99.96%	99.93%	99.95%	99.94%	0.02s
CNN-based Model [12]	98.7%	98.7%	97.0%	97.8%	0.06s
Ensemble Incremental Learning [1]	98.2%	98.5%	97.0%	97.7%	0.03s
Adaptive Class Incremental Learning [2]	98.8%	98.0%	97.5%	97.3%	0.04s
Active Incremental Learning [3]	98.6%	98.8%	97.2%	97.0%	0.05s

Table 7: Comparison results

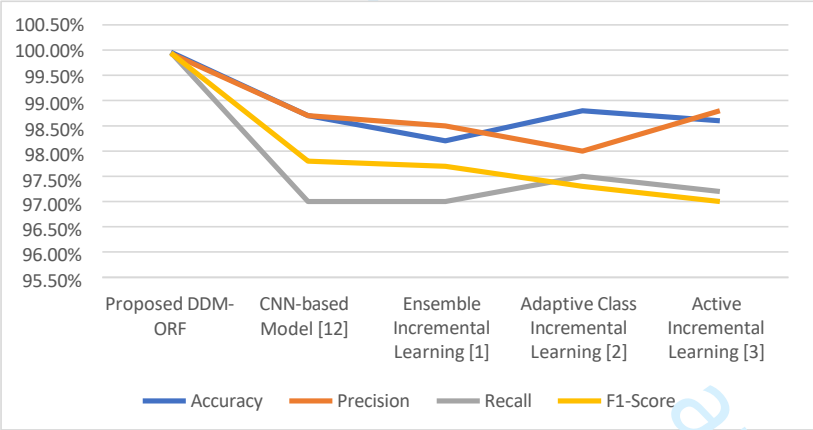


Figure 10: Comparison results

The proposed DDM-ORF method outperformed the other methods in terms of accuracy, precision, recall, and F1-score. Additionally, the processing time of DDM-ORF was significantly lower, highlighting its efficiency in real-time applications.

- **Proposed DDM-ORF**

High Accuracy, Precision, Recall, and F1-Score: The DDM-ORF's high performance metrics suggest that it is highly effective at detecting and adapting to concept drifts in real-time. This can

be attributed to the dynamic updating of the random forest classifier and the efficient handling of new data instances, which allows the model to quickly adapt to changes in the data distribution.

Low Processing Time: The efficient incremental update mechanism of the ORF algorithm, combined with the quick detection of drifts using DDM, contributes to the low processing time. This efficiency makes it suitable for real-time applications where quick response is critical.

- **CNN-based Model [12]**

High Accuracy and Precision but Lower Recall and F1-Score: While the CNN-based model performs well in terms of accuracy and precision, its recall and F1-score are slightly lower, indicating that it may not be as effective in identifying all instances of anomalies or suspicious traffic. This could be due to the static nature of the CNN architecture, which might not adapt as quickly to changes in data patterns as the DDM-ORF.

Higher Processing Time: The deeper architecture of the CNN, with multiple convolutional, pooling, and fully connected layers, along with dropout regularization, increases the computational complexity and processing time compared to the DDM-ORF.

- **Ensemble Incremental Learning [1]**

Good Performance but Slightly Lower than DDM-ORF: The ensemble approach generally offers robustness by combining multiple models, but it might not adapt as swiftly to concept drifts as the DDM-ORF, resulting in slightly lower performance metrics.

Moderate Processing Time: The need to update multiple classifiers in the ensemble leads to a moderate processing time, higher than DDM-ORF but lower than the CNN-based model.

- **Adaptive Class Incremental Learning [2]**

High Accuracy with Some Trade-offs: This method shows good accuracy and recall but slightly lower precision and F1-score, indicating some issues with false positives. Its design for IoT environments might contribute to its solid performance, though it may struggle with imbalanced datasets or require substantial labeled data.

Moderate Processing Time: The approach involves incremental updates that, while efficient, do not match the low processing time of DDM-ORF due to the additional steps required for handling class increments and updating the neural network structure.

- **Active Incremental Learning [3]:**

Good but Not Top Performance: While maintaining good accuracy and precision, the reliance on active learning and human expertise might limit its scalability and adaptability, resulting in slightly lower metrics compared to the DDM-ORF.

Higher Processing Time: The interactive nature of active learning, which involves periodic human intervention, contributes to a higher processing time, making it less suitable for real-time applications compared to the fully automated DDM-ORF.

These results demonstrate the effectiveness of DDM-ORF in maintaining high accuracy and efficiency while handling concept drift and online incremental learning, making it a robust choice for real-time intrusion detection systems.

5.2 Experiment 2

In this evaluation, we analyze the DDM-ORF model predictions to determine and find some insights. Our objective is to identify the top IP addresses responsible for launching attacks and the top IP addresses that were targeted by attacks. Additionally, we aim to determine the ports used for launching attacks and the most frequently attacked ports.

As indicated in Table 4, the MAWILab dataset features a "taxonomy" field that offers a comprehensive classification of each recorded event. This taxonomy can be segregated into two primary categories, anomalous and suspicious.

Our evaluation dataset comprises more than 1,873,545 events. Following the execution of our streaming job and the deduplication operation via Spark Structured Streaming to eliminate duplicate records, we were left with a total of 560,808 events. The subsequent figure illustrates how these events are distributed.

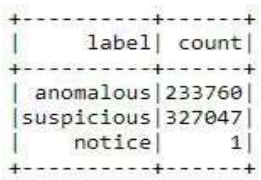


Figure 11: Labels distribution

According to our analysis, anomalous events account for more than 41.68% of the evaluated dataset, while suspicious events account for 58.31%. The top 10 taxonomies found in the dataset, as shown in Table 8, account for 92.15% of the total taxonomies found, with a total of 516821. Other types of attacks such as DoS and DDoS account for only 1.74% of the anomalous traffic. Table 9 displays the top 5 taxonomies found in both anomalous and suspicious traffic. The top 5 anomalous taxonomies, which make up 81.97% of all anomalous events, and the top 5 suspicious taxonomies, with a total of 233926 events, representing 71.52% of all suspicious events.

Taxonomy	**Count**	**Taxonomy**	**Count**
Multi. Points	157018	ntscUDPUDPrp (Network Scan UDP)	24812
HTTP	151866	Ptmpla (HTTP)	22935
Alpha flow	52314	Mptpla (HTTP)	21655
ntscSYNt (Network Scan TCP)	31996	Network scan TCP	19337
Network scan UDP	28981	ntscSYNt139445 (Network scan TCP)	5907

Table 8: Top 10 Taxonomies

Taxonomy (anomalous)	**Count**	**Taxonomy (suspicious)**	**Count**
Multi. points	88429	HTTP	84803
Http	67063	Multi. points	68589
ntscSYNt	14773	Alpha flow	39887
Alpha flow	12427	Network scan UDP	21205
Network scan TCP	8508	ntscUDPUDPrp (network Scan UDP)	19442

Table 9: Top 5 anomalous and suspicious taxonomies

	Destination ports	Count	Source ports	Count
Anomalous	0 (unknown)	123093	0 (unknown)	100656
	80	33109	80	65368
	443	11468	443	18569
	53	11393	53	9531
	22	3809	6000	5906
Suspicious	0 (unknown)	138370	0 (Unknown)	176086
	80	48909	80	55186
	53	25811	53	16228
	443	18217	443	15271
	23	12537	22	2080

Table 10: Anomalous and Suspicious most targeted and used ports

Table 10 shows the most targeted destination ports and the most used source ports for both anomalous and suspicious events. We notice that in the analyzed dataset, the most frequently utilized ports for both anomalous and suspicious traffic were unknown ports, with port 80 (HTTP), 53 (DNS), and 443 (HTTPS) following closely behind. Even though IP addresses aren't always a true source of traffic since they can be masked or changed, they are still used to identify sources of attacks or at least the country of the original attack.

We list in Table 11 the top IP addresses that were the source of anomalous and suspicious activities as well as the top IP addresses that were the destination of such activities.Most of anomalous/suspicious events were originated from unknown sources followed by 0.0.0.0 address which are also from unknown source.

	Destination IP	Count	Source IP	Count
Anomalous	Unknown	87198	Unknown	83334
	0.0.0.0	232	0.0.0.0	409
	193.42.178.137	22	172.20.32.73	42
	193.42.178.130	21	172.20.32.48	40
	10.5.115.115	20	94.164.147.39	15
Suspicious	Unknown	112614	Unknown	91559
	0.0.0.0	1766	0.0.0.0	1278
	10.64.17.12	88	29.15.106.164	55
	10.5.115.115	61	172.20.32.48	54
	10.64.169.9	37	138.131.183.14	50

Table 11: Anomalous and Suspicious most emitting and receiving IP addresses

Our DDM-ORF proposed approach provides multi-class classification (suspicious and anomalous). This enables the detection system to not only detect whether an intrusion attempt is occurring but also to provide insights into the type of attack being attempted. By categorizing intrusion attempts, the system can provide insights into the frequency and type of attacks being attempted, allowing security professionals to better understand and respond to threats.

5.3 Experiment 3

This experiment concerns Apache Spark evaluation. Authors in [35] presented significant metrics for assessing the performance of Apache Spark streaming. As our work involves the ongoing collection of data from the Fukuda Lab, we rely on two metrics to measure performance: Processed Records per Second and Input Rows per Second. These metrics are described in the following table along with the corresponding results.

Metric	Description	Results
Input Rate	Describes how many rows were loaded persecond.	560808 Event/Second
Processing Rate	Describes how many rows were processedper second.	55175 Event/Second

Table 12: Apache Spark evaluation

Our system collected more than 560,808 rows of data, with each file containing an average of 163 rows. The system was able to process these records at a rate of 55,175 records per second and collected all files in a single batch. This high processing rate was made possible by setting the maxFilesPerTrigger parameter to its maximum value, which allowed for the collection of all files at once, rather than limiting the number of files collected at a time.

The performance of our Spark Cluster is influenced by two key factors. The first factor is the size of the cluster, which can affect the system's ability to process data depending on the workload. Allocating more processing power to the cluster generally results in the system being able to process more data. The second factor is the rate at which data is incoming. If the rate exceeds the system's processing capacity, it can cause a bottleneck and require intervention to restrict the input rate.

In general, our proposed DDM-ORF system achieves excellent classification results and processing speed, which are suitable for real-world scenarios and can be further improved by adding more machines to the cluster. Additionally, Apache Spark Structured Streaming provides crucial features for handling streaming data, such as dynamic modification of data structure, filling missing fields, and removing duplicate records. Moreover, Apache Spark is fault-tolerant, distributed, and capable of reading and writing data from various sources in various formats.
In comparison with similar contributions described in the related work section, our DDM-ORF proposed contribution is tested on a large-scale dataset, suitable for heterogenous data, based on online learning approach and provides multi-class classification with very good accuracy.

6. CONCLUSION

This paper proposes a novel DDM-ORF model to detect intrusions based on concept drift detection and online incremental learning. The model uses Drif Detection Method for drift detection and the Online Random Forest algorithm for the incremental learning. Incremental learning algorithms are designed to continuously update the model parameters as new data becomes available and can adapt to changes in the data distribution over time. This makes them well-suited for applications where the data is dynamic and non-stationary. However, it's worth noting that incremental learning

is more computationally complex compared to traditional machine learning, and requires additional resources for training and monitoring the model. Additionally, incremental learning algorithms requires more data to achieve better accuracy levels than traditional machine learning approaches, as they need to continuously update the model parameters to account for changes in the data distribution.

In our contribution, we prioritize the key features of scalability, availability, and fault-tolerance by choosing Apache Spark as our distributed computing framework. This enables us to process large volumes of data simultaneously across multiple machines. To further strengthen our system, we take advantage of Microsoft Azure Cloud's auto-scaling capabilities. This enables us to dynamically adjust the size of our cluster to accommodate changes in workload, without any downtime or interruption of service. Additionally, Microsoft Azure Cloud ensures high availability of our cluster by providing redundancy and failover mechanisms that prevent data loss and maintain system accessibility even in the event of hardware failures or network outages.

Our proposed contribution, DDM-ORF, utilizes an online learning approach and has been tested on a large-scale dataset. It is suitable for heterogenous dataset and provides multi-class classification with high accuracy.

From another side, DDM only considers changes in the frequency of class labels and may not be effective in detecting more complex types of concept drifts. One perspective to address these limitations is to explore other concept drift detection algorithms, such as those based on clustering, density estimation, or change-point detection. Besides, one limitation of online incremental random forest is that it requires more memory to maintain a large number of trees as new data arrives. It can suffer from overfitting if the size of the forest is not carefully controlled. One perspective to overcome this limitation is to use ensemble pruning techniques that can dynamically remove unnecessary trees while retaining the accuracy of the model. Another perspective is to explore new algorithms that can effectively balance the trade-off between accuracy and memory usage, such as compressed or compact random forests.

ETHICAL STATEMENTS

I. The authors declare that this manuscript is original, has not been published before and is not currently being considered for publication elsewhere. II. The authors declare that they have no funding. III. The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper. IV. The authors have read and agree to the Terms and Conditions of the journal.

DATA AVAILABILITY

The datasets generated during and/or analyzed during the current study are available from the authors.

References

[1] X. Yuan, R. Wang, Y. Zhuang, K. Zhu, and J. Hao, "A Concept Drift Based Ensemble Incremental Learning Approach for Intrusion Detection," 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Halifax, NS, Canada, 2018, pp. 350-357, doi: 10.1109/Cybermatics_2018.2018.00087.

[2] Q. Liu, Y. Zhang, W. Zhou, X. Jiang, W. Zhou, and M. Zhou, "Adaptive Class Incremental Learning-Based IoT Intrusion Detection System," Computer Engineering, vol. 49, no. 2, pp. 169-174, 2023.

[3] Z. Sun, G. Ran, and Z. Jin, "Intrusion detection method based on active incremental learning in industrial internet of things environment," Journal on Internet of Things, vol. 4, no. 2, pp. 99-111, 2022.

[4] Kuppa and N.-A. Le-Khac, "Learn to adapt: Robust drift detection in security domain," Computers and Electrical Engineering, vol. 102, p. 108239, 2022, doi: 10.1016/j.compeleceng.2022.108239.

[5] Z. Wu, P. Gao, L. Cui, and J. Chen, "An Incremental Learning Method Based on Dynamic Ensemble RVM for Intrusion Detection," IEEE Transactions on Network and Service Management, vol. 19, no. 1, pp. 671-685, 2022.

[6] E. Mahdavi, A. Fanian, A. Mirzaei, and Z. Taghiyarrenani, "ITL-IDS: Incremental Transfer Learning for Intrusion Detection Systems," Knowledge-Based Systems, vol. 253, p. 109542, 2022, doi: 10.1016/j.knosys.2022.109542.

[7] G. Folino, F. S. Pisani, and L. Pontieri, "A GP-based ensemble classification framework for time-changing streams of intrusion detection data," Soft Computing, 2020.

[8] S. Dwibedi, M. Pujari, and W. Sun, "A Comparative Study on Contemporary Intrusion Detection Datasets for Machine Learning Research," 2020 IEEE International Conference on Intelligence and Security Informatics (ISI), 2020.

[9] Guarino, G. Bovenzi, D. Di Monda, G. Aceto, D. Ciuonzo, and A. Pescapé, "On the use of Machine Learning Approaches for the Early Classification in Network Intrusion Detection," 2022 IEEE International Symposium on Measurements & Networking (M&N), 2022.

[10] E. Nugroho, T. Djatna, I. S. Sitanggang, A. Buono, and I. Hermadi, "A Review of Intrusion Detection System in IoT with Machine Learning Approach: Current and Future Research," 6th International Conference on Science in Information Technology (ICSITech), 2020.

[11] Karthika, S. Loganathan, and M. Vanathi, "A Hybrid Machine Learning Based Feature Selection Technique for Attack Detection in NIDS.

[12] R. Dhahbi and F. Jemili, "A Deep Learning Approach for Intrusion Detection," 2021 IEEE 23rd International Conference on High Performance Computing & Communications (HPCC), 2021, pp. 1-8, doi: 10.1109/HPCC-SmartCity-DSS51687.2021.00033.

[13] Y. Kamel, F. Jemili, and R. Meddeb, "Ensemble learning based big data classification for intrusion detection," in 22nd International Conference on Intelligent Systems Design and Applications, Springer, 2022, pp. 1-8.

[14] F. Jemili, "Towards Data Fusion-based Big Data Analytics for Intrusion Detection," Journal of Information & Telecommunication, 2023, doi: 10.1080/24751839.2023.2214976.

[15] Abid and F. Jemili, "Intrusion Detection based on Graph oriented Big Data Analytics," in KES-2020 24th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, 2020, pp. 448-457, doi: 10.1016/j.procs.2020.08.059.

[16] M. Hafsa and F. Jemili, "Comparative Study between Big Data Analysis Techniques in Intrusion Detection," Big Data and Cognitive Computing, vol. 3, no. 1, pp. 1-12, Dec. 2018, doi: 10.3390/bdcc3010001.

[17] F. Jemili, "Intelligent intrusion detection based on fuzzy Big Data classification," Cluster Computing, 2022, doi: 10.1007/s10586-022-03769-y.

[18] G. D'Angelo, F. Palmieri, and A. Robustelli, "Effectiveness of Video-Classification in Android Malware Detection Through API-Streams and CNN-LSTM Autoencoders," in 5th International Symposium on Mobile Internet Security (MobiSec), 2021, pp. 171-194.

[19] R. Meddeb, F. Jemili, B. Triki, and O. Korbaa, "A deep learning based intrusion detection approach for mobile ad-hoc network," Soft Computing, 2023. [DOI: 10.1007/s00500-023-08324-4]

[20] M. Coccia, S. Roshani, and M. Mosleh, "Scientific Developments and New Technological Trajectories in Sensor Research," Sensors, vol. 21, no. 23, p. 7803, 2021, doi: 10.3390/s21237803.

[21] S. Pamarthi and R. Narmadha, "Literature review on network security in Wireless Mobile Ad-hoc Network for IoT applications: network attacks and detection mechanisms," International Journal of Intelligent Unmanned Systems, vol. 10, no. 4, pp. 482-506, 2022.

[22] Hasan et al., "Forensic analysis of blackhole attack in wireless sensor networks/internet of things," Applied Sciences, vol. 12, no. 22, p. 11442, 2022, doi: 10.3390/app122211442.

[23] Abid, F. Jemili, and O. Korbaa, "Distributed architecture of an Intrusion Detection System in Industrial Control Systems," in ICCCI 2022 14th International Conference on Computational Collective Intelligence, 2022.

[24] M. Coccia, S. Roshani, and M. Mosleh, "Evolution of Sensor Research for Clarifying the Dynamics and Properties of Future Directions," Sensors, vol. 22, no. 23, p. 9419, 2022, doi: 10.3390/s22239419.

[25] Wang and R. Jones, "Big data analytics for network intrusion detection: A survey," International Journal of Networks and Communications, vol. 7, no. 1, pp. 24-31, 2017.

[26] Z. Sultan and M. İskefiyeli, "Anomaly-based intrusion detection from network flow features using variational autoencoder," IEEE Access, vol. 8, pp. 108346-108358, 2020.

[27] Y. Zhou, G. Cheng, S. Jiang, and M. Dai, "Building an efficient intrusion detection system based on feature selection and ensemble classifier," Computer Networks, vol. 174, p. 107247, 2020.

[28] A. Salih and A. M. Abdulazeez, "Evaluation of classification algorithms for intrusion detection system: A review," Journal of Soft Computing and Data Mining, vol. 2, no. 1, pp. 31-40, 2021.

[29] Shaukat, S. Luo, V. Varadharajan, I. A. Hameed, S. Chen, D. Liu, and J. Li, "Performance comparison and current challenges of using machine learning techniques in cybersecurity," Energies, vol. 13, no. 10, p. 2509, 2020.

[30] P. Singh and V. Ranga, "Attack and intrusion detection in cloud computing using an ensemble learning approach," International Journal of Information Technology, vol. 13, no. 2, pp. 565-571, 2021.

[31] A. Tama, M. Comuzzi, and K.-H. Rhee, "Tse-ids: A two-stage classifier ensemble for intelligent anomaly-based intrusion detection system," IEEE Access, vol. 7, pp. 94497-94507, 2019.

[32] Thakkar and R. Lohiya, "A review on machine learning and deep learning perspectives of IDS for IoT: Recent updates, security issues, and challenges," Archives of Computational Methods in Engineering, vol. 28, no. 4, pp. 3211-3243, 2021.

[33] J. N. van Rijn, G. Holmes, B. Pfahringer, and J. Vanschoren, "The online performance estimation framework: Heterogeneous ensemble learning for data streams," Machine Learning, vol. 107, no. 1, pp. 149-176, 2018.

[34] R. Panigrahi and S. Borah, "A detailed analysis of CICIDS2017 dataset for designing intrusion detection systems," International Journal of Engineering & Technology, vol. 7, no. 3.24, pp. 479-482, 2018.

[35] T. Ivanov and J. Taaffe, "Exploratory Analysis of Spark Structured Streaming," in International Conference on Performance Engineering, Berlin, 2018.

[36] I. Salah, K. Jouini, and O. Korbaa, "Augmentation-based ensemble learning for stance and fake news detection," in Advances in Computational Collective Intelligence - 14th International Conference, ICCCI 2022, Hammamet, Tunisia, September 28-30, 2022, Proceedings, vol. 1653 of Communications in Computer and Information Science, Springer, 2022, pp. 29-41.

[37] O. Abdel Wahab, "Intrusion Detection in the IoT Under Data and Concept Drifts: Online Deep Learning Approach," in IEEE Internet of Things Journal, vol. 9, no. 20, pp. 19706-19716, 15 Oct.15, 2022, doi: 10.1109/JIOT.2022.3167005.

.

Q3.1 Aggregates Selection in Replicated Document-Oriented Databases

Khaled Jouini

Journal of Information Science and Engineering (J INF SCI ENG). 2022.
ISSN : 1016-2364, DOI : 10.6688/JISE.20220338(2).0012
JCR IF (2022) : 1.1

Lien : https://jise.iis.sinica.edu.tw/JISESearch/pages/View/PaperView.jsf?keyId=185_2505
SJR best quartile : Q3, SJR : 0.21

Journal of Information Science and Engineering

COUNTRY

Taiwan

Universities and research institutions in Taiwan

Media Ranking in Taiwan

SUBJECT AREA AND CATEGORY

Computer Science
- Computational Theory and Mathematics
- Hardware and Architecture
- Human-Computer Interaction
- Software

Social Sciences
- Library and Information Sciences

PUBLISHER

Institute of Information Science

H-INDEX

42

PUBLICATION TYPE

Journals

ISSN

10162364

COVERAGE

1993-1994, 1996-2023

INFORMATION

Homepage

How to publish in this journal

tshsu@iis.sinica.edu.tw

SCOPE

The Journal of Information Science and Engineering is dedicated to the dissemination of information on computer science, computer engineering, and computer systems. This journal encourages articles on original research in the areas of computer hardware, software, man-machine interface, theory and applications. tutorial papers in the above-mentioned areas, and state-of-the-art papers on various aspects of computer systems and applications.

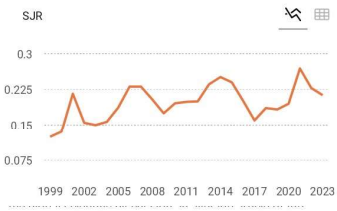
Join the conversation about this journal

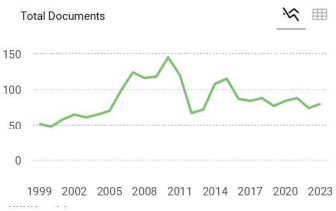
Quartiles

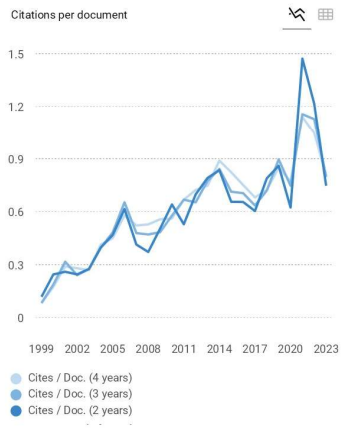
FIND SIMILAR JOURNALS

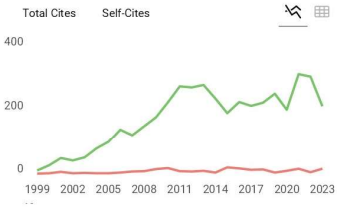
options

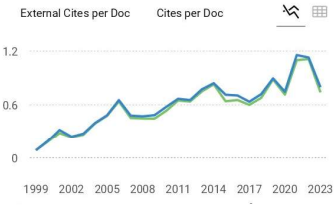
1	2	3	4	5
ACM Computing Surveys	**SN Computer Science**	**Computer Science Review**	**Array**	**Computing and Informatics**
USA	DEU	IRL	NLD	SVK

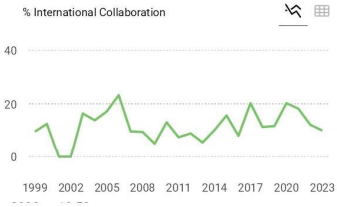


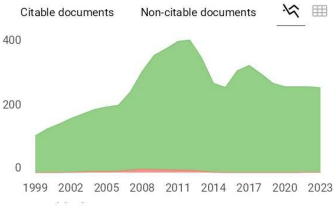


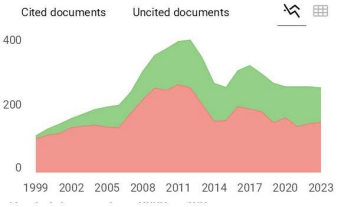


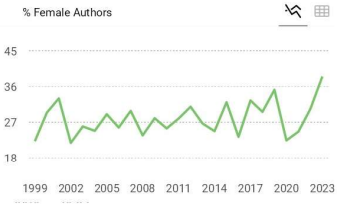


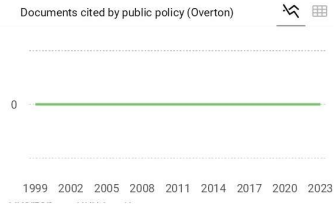


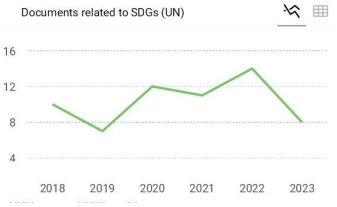












← Show this widget in your own website

Just copy the code below and paste within your html code:

<a href="https://www.scim

SCImago Graphica

Explore, visually communicate and make sense of data with our **new data visualization tool.**

Metrics based on Scopus® data as of March 2024

2022 Journal Performance Data for: JOURNAL OF INFORMATION SCIENCE AND ENGINEERING

ISSN	EISSN
1016-2364	N/A

JCR ABBREVIATION	ISO ABBREVIATION
J INF SCI ENG	J. Inf. Sci. Eng.

Journal Information

EDITION	CATEGORY
Science Citation Index Expanded (SCIE)	COMPUTER SCIENCE, INFORMATION SYSTEMS - SCIE

LANGUAGES	REGION	1ST ELECTRONIC JCR YEAR
English	TAIWAN	2000

Publisher Information

PUBLISHER	ADDRESS	PUBLICATION FREQUENCY
INST INFORMATION SCIENCE	ACADEMIA SINICA, TAIPEI 115, TAIWAN	6 issues/year

Journal Impact Factor

The Journal Impact Factor (JIF) is a journal-level metric calculated from data indexed in the Web of Science Core Collection. It should be used with careful attention to the many factors that influence citation rates, such as the volume of publication and citations characteristics of the subject area and type of journal. The Journal Impact Factor can complement expert opinion and informed peer review. In the case of academic evaluation for tenure, it is inappropriate to use a journal-level metric as a proxy measure for individual researchers, institutions, or articles. Learn more

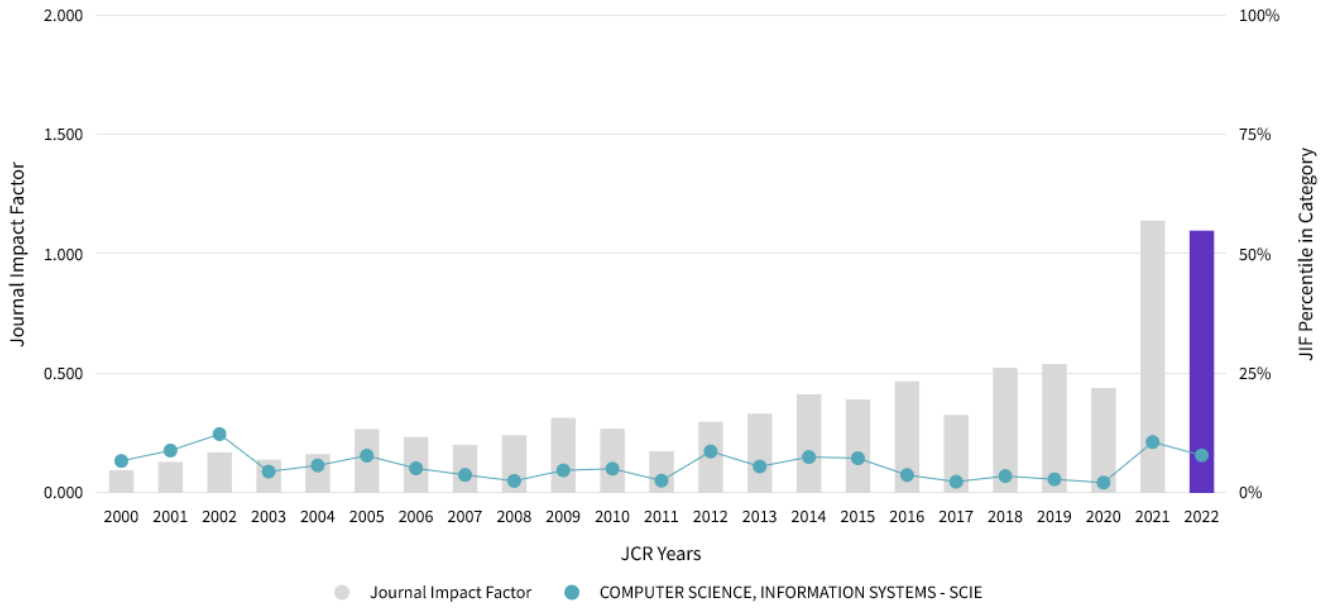
2022 JOURNAL IMPACT FACTOR

1.1

2022 JOURNAL IMPACT FACTOR WITHOUT SELF CITATIONS

1.1

Journal Impact Factor Trend 2022



JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 38, 479-496 (2022)
DOI: 10.6688/JISE.202203_38(2).0012

Aggregates Selection in Replicated Document-Oriented Databases

KHALED JOUINI
MARS Research Lab LR17ES05
ISITCom, University of Sousse
H. Sousse, 4011 Tunisia
E-mail: khaled.jouini@isitc.u-sousse.tn

Document-stores leverage the flexibility of structured documents to pack closely related data within a single autonomous *aggregate* (*i.e.* document). Selecting an appropriate set of aggregates for a document database is a non-trivial task since: (i) there are no clear-cut transformation rules from a conceptual design to a document design; (ii) a large space of design options must often be considered; and (iii) most importantly, it is difficult, if not impossible, to find out a single set of aggregates suitable for all data access patterns.

In a previous work, we proposed *distorted replicas*: a replication scheme that leverages ubiquitous replication in document-stores and restructures replicated data in different ways to better cope with the heterogeneity of data access patterns. In this paper, we tackle the problem of aggregates selection and replication in an integrated manner. In particular, given a database with a replication factor of C and a workload W, we propose novel cost-driven techniques allowing to: (i) determine the most interesting aggregates; and (ii) pack the most interesting aggregates into C disjoint and complete subsets in such a way that the execution time of W is minimized. Experimental results obtained over two real-world workloads showed that distorted replicas allow to run queries up to tens of times faster than state-of-the-art approaches.

Keywords: logical & physical design, aggregate data model, replication, 0-1 MKP, document-stores

1. INTRODUCTION

NoSQL systems rise has been driven by the desire to store data on large clusters of commodity servers and to provide horizontal scalability, high availability, and high throughput for write/read operations [1]. Document-stores leverage the flexibility of structured documents (*e.g.* JSON) to pack closely related data in a single autonomous document or *aggregate*, rather than having them scattered across several tables as in the relational model. By doing so, document-stores manipulate related data in a single database operation and avoid cross-nodes writes and joins, which are prohibitive in highly distributed environments [2]. Fig. 1 depicts a slightly modified JSON-formatted document from the archives of the DBLP bibliography [3] and illustrates the key differences between the aggregate data model and the relational data model.

Received March 16, 2020; revised August 26, 2020; accepted October 5, 2020.
Communicated by Reynold C.K. Cheng.

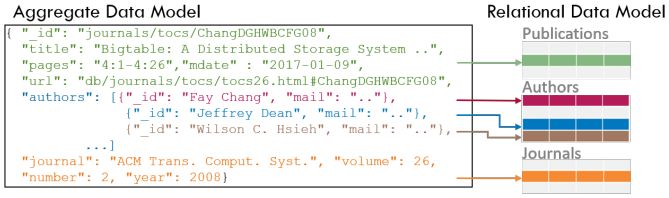


Fig. 1. Aggregate data model vs. Relational data model.

A key challenge in document-stores is how to model documents to meet the needs of applications in terms of performance and access patterns. Several causes make the selection of an appropriate set of aggregates a non-trivial task. First, there are no clear-cut transformation rules from a conceptual design to a document design, as efficient as the normalization process of relational databases [4]. Second, a wide range of alternative design options must often be considered (Fig. 2). When the number of entities and relationships is large, this may easily lead to a combinatorial explosion of alternative candidate schemas [4]. Third, a judicious modeling choice depends entirely on how we tend to manipulate data [1], and hence, must be *cost-driven* and influenced by the workload experienced by the system (*i.e. workload-aware*) [5]. Last and most importantly, it is commonly accepted that it is difficult, if not impossible, to find out a single set of aggregates suitable for all queries [4].

Replication is ubiquitous in NoSQL systems. In a previous work [6], we proposed a new replication scheme called *distorted replicas*. The main idea behind distorted replicas is to restructure replicated documents in different ways to better cope with the unavoidable heterogeneity of data access patterns. The idea of organizing replicated data in different ways was first introduced in [7] and applied to relational databases hosted on servers with mirrored disks. The idea was next applied to replicated blocks in distributed file systems [8]. We think that restructuring replicated data is much more of a central aspect for aggregate-oriented databases than it is for relational databases, since most applications will have to deal with queries that do not fit well with the aggregate structure.

The problem we tackle in this paper is as follows. Given a database with a replication factor of C (*i.e.* a database replicated C times) and an incoming query workload W, we have to determine C subsets of complete and disjoint aggregates that optimize W (*i.e.* that minimize the execution time of W). While there has been work in the area of document-oriented database design [4,9,10], we are not aware of any work that addresses the problem of aggregates selection and replication in an integrated manner. To deal with the logical and physical design challenges triggered by distorted replicas, we make the following key contributions: (i) we show how to identify interesting aggregates and how to assign them an *interestingness* value; (ii) we map the problem of aggregates selection to a *0-1 multiple knapsack problem* and solve it using a *branch and bound technique*; and (iii) we evaluate our approach on top of MongoDB using two real-world datasets: DBLP [3] and TPC-H [11]. The obtained results show that distorted replicas can execute queries up to tens of times faster than state-of-the-art approaches.

The remainder of this paper is organized as follows. Section 2 briefly reviews the main concepts related to document-stores. Section 3 discusses our workload-aware, cost-based algorithm for aggregates selection. Section 4 presents related works. Section 5

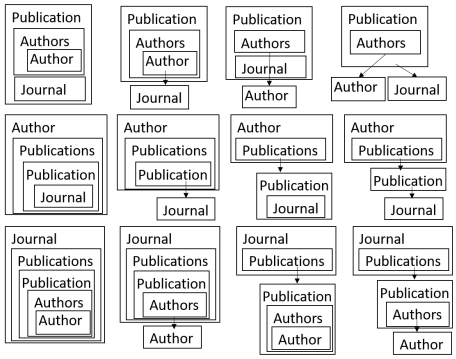


Fig. 2. Alternative schemas for the DBLP dataset; Nested rectangles represent embedded documents; Arrows represent references between aggregates.

gives an experimental study of distorted replicas performance. Section 6 discusses the main results and findings. Section 7 concludes the paper.

2. ANATOMY OF A DOCUMENT-STORE

There exists a wide range of academic and commercial document-stores, each with some features that may not exist in others. In the sequel, we use MongoDB as a representative of the feature set but also reference other document-store systems.

2.1 Document Modeling: Challenges and Considerations

2.1.1 Read-overhead

One of the most critical document-modeling choices is how to represent relationships between data: with references (*i.e. normalized data model*) or with embedded documents (*i.e. denormalized data model*). References represent relationships between data by including a link from one document to another, just as in the relational model. The normalized data model strives for a single copy of the data, minimizing redundancy and favoring consistency [4]. However, if related data is stored in separate servers, joins and writes may be prohibitively slow.

Embedded documents represent relationships by storing related objects in a single aggregate and hence, avoid cross-nodes joins and writes. Aggregates are useful in that they pack into one document, objects that are expected to be accessed together. However, there are many use cases where objects or fields need to be accessed individually. When a field needs to be accessed individually, not only that field's value is loaded in the memory hierarchy, but also all the data within the same aggregate. Loading large amounts of data irrelevant for a given query may seriously waste main memory, disk, and network bandwidths and increase the number of CPU cycles wasted in waiting for data loading [12]. The introduced *read-overhead* is one of the most important downsides of aggregate-oriented data models.

2.1.2 Aggregate roots

Objects in an aggregate are bound together by a root object, known as the *aggregate root* [1]. In most cases, there exist many root candidates. Grouping objects by one of the roots may help with some data interactions but is necessarily an obstacle for many others. As reported in [1], the entire aggregate orientation approach works well only when data access is aligned with aggregate roots. If data is accessed in a different way, the whole system performance may be substantially impacted. Consider the DBLP example of Fig. 1 and the relationship between authors and publications. Some queries will require to access authors whenever they access a publication; this fits in well with combining a publication with its authors into a single aggregate that can be stored and accessed as a unit. Other queries, however, will require to access the history of publications whenever they access an author. In such a case, it would be necessary to dig into every aggregate in the database. As aptly stated in [1], we can reduce this burden by building secondary indexes, but *we're still working against the aggregate structure*. Relational databases have an advantage here as they allow to slice and dice data in different ways for different queries.

In the following, an aggregate a is represented by a pair (r, E), where $a.E$ is the set of entities embedded within a and $a.r \in E$ is the root of a. An aggregate formed by a single entity is said to be *atomic*. To benefit from the efficiency of bitwise operations in our algorithms, $a.E$ is represented by a bitmap of $\mid \mathbb{E} \mid$ bits, where $\mathbb{E}$ is the set of modeled entities. The ith bit of the bitmap is set to 1 if the ith modeled entity is embedded within the aggregate, and to 0 otherwise.

2.2 Replication

Replication is the process of maintaining different replicas of the same data on different servers. The primary purpose of replication is to enhance availability and fault-tolerance by providing multiple paths to redundant data. Replication can also be used to increase: (i) I/O throughput by distributing requests across servers; and (ii) data locality by allowing a client application to access data from the closest server.

A set of servers maintaining replicas of the same data (sub-)set is called a *Replica Set* (or *RS*) in MongoDB. A replica set is composed of one master node, called *primary*, and a set of slave nodes, called *secondaries*. The primary node is the only member in a replica set that receives writes. When the primary receives a write request, it updates its data set and records the write in the operations Log (*i.e. opLog*). Secondary nodes periodically import the opLog and apply all changes to their local replicated collections in such a way that they reflect the master collections [13]. As in most NoSQL systems, replication in MongoDB is by default asynchronous: (i) there may exist a delay between the occurrence of an operation on the primary and its application on a secondary (*i.e. replication lag*); and (ii) the client application does not have to wait for the completion of a write on slaves.

2.3 Sharding

Sharding is similar to horizontal partitioning in RDBMSs. It consists in splitting a data set according to a given field, called the *shard key*. The resulting data subsets are called *chunks* and are hosted on multiple separate servers, called *shards*. Each shard is an *independent database* having its own subset of data stored on its own local disks. In MongoDB, each shard can be a complete replica set. A prominent concern in sharding

is to balance the load between shards. Typically, when a chunk grows beyond a given size, it is split causing an increase in the number of chunks held by the server. If the chunk distribution becomes uneven, some chunks are migrated from the shard that has the largest number of chunks to the one with the least number of chunks, until the cluster is rebalanced. A similar process occurs when a new shard is added to the cluster.

3. WORKLOAD-AWARE, COST-BASED AGGREGATES SELECTION

3.1 Overview

The main idea behind distorted replicas [6] is to restructure replicated data in different ways to better cope with the unavoidable heterogeneity of data access patterns. In our work, data restructuring is materialized by: (i) converting a reference between two aggregates into an embedded document and inversely, and (ii) reorganizing data according to different aggregate roots. Note here that restructuring aggregates according to new roots is not tedious as document-stores provide optimized operators allowing to promote an entity from "embedded" to "root" (*e.g.* `$replaceRoot` in MongoDB). We should also note that in our work data is only reorganized locally, *i.e.* within the same replica set (shard). Accordingly, even if references are used, we do not have to perform costly cross-nodes joins to reconstruct an aggregate.

We assume that we are given a database D with a replication factor C and a representative workload W, for which we need to recommend aggregates and their "distorted" replicas. The workload can be obtained when *migrating from RDBMS to NoSQL* as in [14, 15] or *by processing the Log of the database* as in [8, 16]. Our goal is to find C disjoint subsets of aggregates, such that the performance of W is optimized, subject to two constraints: (1) *Disjointness constraint*: aggregates within the same subset are disjoint, which means that within a replica set member, each modeled entity appears in at most one aggregate; (2) *Restorability constraint*: while not physically identical, replicas have to be logically identical. This means that all entities must appear within each replica set member.

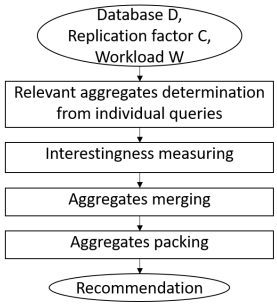


Fig. 3. Aggregates selection: key steps.

The key steps of our solution, pictured in Fig. 3, are in the spirit of S. Chaudhuri & V. Narasayya work on index, materialized views, and horizontal/vertical partitions selection in RDBMSs, summarized in the VLDB ten-year best paper award of S. Chaudhuri [16]. For simplicity of exposition, we retain the terminology used in [16] wherever applicable.

As illustrated in Fig. 2, an aggregate-oriented database designer is faced with a plethora of alternative design options: what entity is the owner of a relationship (*i.e.* what entity should embed a relationship), which relationships to denormalize, to which depth (*i.e.* subsequent relationships), *etc.*. Enumerating all possible combinations of embedding and referencing, becomes exponential as the number of entities and relationships increases [4]. To limit the space of the considered aggregates, we first restrict ourselves to those *relevant* for at least one query in the workload. Each relevant aggregate is then

assigned with a value quantifying its *interestingness*.

An aggregate suitable for one query, may substantially degrade the performance of another query. The goal of the "relevant aggregates merging" step is hence to find additional aggregates, that although are not optimal for any individual query, are useful for multiple queries, and therefore could be optimal for the workload. Given the set of interesting aggregates and a replica set of C members, the "aggregates selection" step aims to select C subsets of complete and disjoint aggregates so that the total interestingness of the replica set is a maximum.

As in [5,8,16], we assume that there is a function $Cost(q,a)$ that returns the optimizer estimated cost of a query q when q is answered using an aggregate a. As stated in [16], many database systems support the necessary interfaces to answer such "what-if" questions. In the absence of such interfaces, we could follow the same approach as [5, 8] and estimate $Cost(q,a)$ by the footprint of q, *i.e.* by the total number of bytes read (*i.e.* consumed) by q. The footprint of q corresponds to: $Size(a) \times Cardinality(a) \times Selectivity(q)$, where $Size(a)$ is the estimated size of a, $Cardinality(a)$ the estimated number of a instances, and $Selectivity(q)$ the percentage of instances that q selects (with 0 meaning no instances and 1 meaning all instances). If no index is used, a full scan is necessary and $Selectivity(q)$ is dropped from the formula. Without any loss of generality, one can supply other $Cost(q,a)$ estimations to our algorithm.

3.2 Determining Relevant Aggregates from Individual Queries

In the following, the term *rootable entities* is used to denote entities in a query that can potentially be good aggregate roots. Rootable entities form the basis for "determining relevant aggregates from individual queries", the first step of our approach (Fig. 3). We define rootable entities as follows. Let $q.E$ be the set of modeled entities referenced in a query $q \in W$. Intuitively, an entity $e \in q.E$ is *rootable* for q, if it is potentially useful to group the entities referenced in q by e (*i.e.* to create an aggregate rooted at e that embeds all of the entities referenced in q). We consider that an entity $e \in q.E$ is rootable for q if there exists a field f of e such that f appears in the `Group By` clause, the `Order By` clause, or in a filter predicate of q. In the absence of at least one entity satisfying one of the above conditions, each $e \in q.E$ is considered as a rootable entity.

Intuitively, an aggregate a is *relevant* for a query q if q is answerable using a, hence, if all of $q.E$'s entities are embedded within a ($q.E \subseteq a.E$). Considering all relevant aggregates for a query is not scalable since, in principle, we would have to consider any aggregate a such that $q.E \subseteq a.E$. To prune the space of relevant aggregates, we only consider aggregates whose roots are rootable entities and approach the task of "relevant aggregates selection" in two steps. From each query $q_i \in W$, we first derive a preliminary set A_i of aggregates relevant for q_i. Each of those aggregates: (i) is rooted at a rootable entity for q_i; and (*ii*) contains exactly all of the entities referenced in q_i (and nothing else). For each $a_j \in A_i$, we next check its relevance for the remaining workload queries. The output of this stage is a relevance matrix $R(q_i,a_j)$, indicating whether or not an aggregate a_j is relevant for a query q_i, *i.e.* $R(q_i,a_j) = 1$ if a_j is relevant for q_i, and 0 otherwise.

Example. Let's consider the following self-explanatory SQL queries.
q_1: *"select a.mail from author a where a.name=.."*
q_2: *"select a .id, count(*) from author a, publication p where p.year=.. group by a.id"*.

a_1=*(author,*{*author*}*)* is a relevant aggregate for q_1. a_2=*(author,*{*author, publication*}*)* and a_3=*(publication,*{*author, publication*}*)* are relevant aggregates for q_2.
As $a_1.r = a_2.r$ and $a_1.E \subseteq a_2.E$, we conclude that a_2 is also relevant for q_1. Accordingly, $R(q_1,a_2)$ is set to 1. In contrast to a_2, a_3 is not relevant for q_1, since $a_3.r \neq a_1.r$. $R(q_1,a_3)$ does not change and remains equal to 0, as shown in the relevance matrix opposite.

	q_1	q_2
a_1	1	0
a_2	1	1
a_3	0	1

△

3.3 Measuring Interestingness of Relevant Aggregates

Our next goal is to define a metric that captures the relative *interestingness* of each relevant aggregate. Such a metric is essential to rank aggregates and pick the most valuable ones in the final replica set. Intuitively, an aggregate is interesting for a workload W, if it speeds up a significant fraction of W's queries, *i.e.* allows to significantly reduce the total cost of W. The relevance matrix is useful for indicating which aggregates are relevant for which queries. However, it cannot be used by its own to determine interestingness as it does not take into account neither the relative importance (*i.e.* footprints) of queries nor the read overhead introduced by answering a query q using an aggregate that embeds entities useless for q (not referenced in q).

Let A be the set of all relevant aggregates, $A_i \subseteq A$ the set of aggregates relevant for a query q_i ($\forall\, a_j \in A$, $R(q_i,a_j) = 1$), $Opt(W)$ the optimal cost of W, and $Opt(q_i)$ the optimal cost of q_i. $Opt(q_i)$ is defined as the lowest achievable cost to answer q_i: $Opt(q_i) = Min_{a_j \in A_i}(Cost(q_i,a_j))$. $Opt(W)$ is the lowest achievable cost of W: $Opt(W) = \sum_{1}^{|W|} Opt(q_i)$. We define $Int(a) \in [0,1]$, the *interestingness* of an aggregate a for a workload W, as the fraction of the cost of all queries in W for which a is relevant. Formally, $Int(a) \rightarrow [0,1]$ is defined as follows: $Int(a) = \frac{\sum_{1}^{|W|} R(q_i,a) \times \frac{Opt(q_i)^2}{Cost(q_i,a)}}{Opt(W)}$.

$Int(a)$ is normalized by the total workload cost to make it comparable. This can be used as well if it is necessary to prune aggregates by discarding from further consideration those whose interestingness is below a predefined threshold.

Example. Consider for simplicity a workload consisting of one query q and 3 relevant aggregates a_1, a_2, a_3, such that $Cost(q,a_1)$ = 10 units, $Cost(q,a_2)$ = 15 units and $Cost(q,a_3)$ = 20 units. The relative interestingness is: $Int(a_1) = 1$, $Int(a_2) = 0.67$ and $Int(a_3) = 0.5$. △

Example. Assume that W consists of 2 queries q_1 and q_2 and that we have 4 relevant aggregates a_1, a_2, a_3, and a_4. Each cell in the following table gives the cost of answering a query q_i using an aggregate a_j (with "-" meaning that a_j is not relevant for q_i).
As illustrated in the table opposite, a_1 is optimal for q_1 and a_2 is optimal for the more expensive query q_2 (having higher optimal cost). a_3 and a_4 allow to answer q_1 and q_2 but introduce a read overhead. The interestingness of each aggregate is: $Int(a_1) = 0.33$, $Int(a_2) = 0.67$, $Int(a_3) = 0.76$ and $Int(a_4) = 0.5$.

	q_1	q_2
a_1	10	-
a_2	-	20
a_3	**15**	**25**
a_4	20	40

Intuitively, a_2 is more interesting than a_1 for the given workload, as it is optimal for a query more expensive than the query for which a_1 is optimal. a_3 is more interesting than a_4 as it allows to answer q_1 and q_2 with a lower read overhead. a_3 is more interesting than a_2 as it allows to answer q_1 without introducing a "high" additional cost for q_2. △

3.4 Merging Pairs of Interesting Aggregates

The *disjointness constraint* states that non-disjoint aggregates cannot be stored within the same replica set member. This is essential to avoid data redundancy and, hence, data inconsistency. Due to the disjointness constraint, a large part of relevant aggregates derived from individual queries will be rejected and we can end up with sub-optimal recommendations for the workload. The intuition behind "aggregates merging" is that merging two mutually exclusive aggregates, called the *parent aggregates*, in one sub-optimal aggregate, called the *merged aggregate*, is in some cases better than retaining one parent and rejecting the other. As the merged aggregate and its parents are mutually exclusive, the merged aggregate: (i) should be usable (*i.e.* relevant) in answering all queries where each of its parents was used; and (ii) the cost of answering queries using it should not be "much higher" than the cost of answering queries using one of its parents.

In our work, we only merge aggregates that meet the two following criteria. The parent aggregates must have: (i) the same root; and (ii) exactly one non-common entity with the merged aggregate. The first condition is necessary to ensure that the merged aggregate is relevant for all queries for which one of its parents is relevant. The second condition ensures that the cost of answering these queries using the merged aggregate is not "much higher" than the cost of answering them using its parents. The merged aggregate is retained only when its interestingness is greater than the lowest interestingness of its parents.

Example. Let $a_3 = (r, \{r,s,t\})$ be a merged aggregate and $a_1 = (r, \{r,s\})$, $a_2 = (r, \{r,t\})$ its parent aggregates. Assume that the cost matrix is as follows.

	q_1	q_2
a_1	10	-
a_2	-	20
a_3	20	40

The interestingness of a_1, a_2 and a_3, is respectively $Int(a_1) = \frac{1}{3}$, $Int(a_2) = \frac{2}{3}$ and $Int(a_3) = \frac{1}{2}$. a_3 is therefore retained.
Suppose now that $Cost(q_1, a_3) = 50$ and $Cost(q_2, a_3) = 100$. In such case, $Int(a3) = \frac{1}{5}$ and a_3 is not retained as retaining a_2 is a better choice (the gain achieved is lower than the loss caused by the read-overhead). △

3.5 Interesting Aggregates Selection as a 0-1 MKP

Given the set I of interesting aggregates and C replica-set members ($C \ll |I|$), our goal is to select C disjoint subsets of I, so that: (i) the total interestingness of the selected aggregates is a maximum; and (ii) the disjointness and the recoverability constraints are met. The aggregates selection problem can be likened to a 0-1 multiple knapsack problem (MKP) which is known to be NP-hard. More precisely, given C knapsacks (replica set members) and N items (interesting aggregates), we have to find binary variables x_{ij}, $i \in \{1..C\}$, $j \in \{1..N\}$, having the following meaning: $x_{ij} = 1$ if aggregate j is assigned to member i, and 0 otherwise. Formally, the problem is stated as follows. Let B be a bitmap with $|\mathbb{E}|$ bits all set to 1:

```
   Input  : items, knapsacks, level, nodeID, bestNode, maxProfit
   Output: Id of the decision-tree node that corresponds to the optimal solution
1  if j < items.length() − 1 then
2      B&B(items, knapsacks, level+1, nodeID + ".0", bestNode, maxProfit)
3  for i = 0; i < knapsacks.length(); i++ do
4      if Hamming_weight(knapsacks[i].bitmap ∧ items[level].bitmap) == 0 then
5          knapsacks[i].profit += items[level].profit
6          knapsacks[i].bitmap = knapsacks[i].bitmap ∨ items[level].bitmap
7          if level < items.length() − 1 then
8              B&B (items, knapsacks, level+1, nodeID + "." + str(i+1), bestNode,
                maxProfit)
9          else if knapsacks[i].profit > maxProfit then
10             maxProfit = knapsacks[i].profit
11             bestNode = nodeID
```

Fig. 4. A Branch&Bound algorithm for aggregates packing.

$$\text{maximize} \sum_{i=1}^{C}\sum_{j=1}^{N} Int(a_j)x_{ij} \qquad \text{subject to,}$$

$$x_{ij} \in \{0,1\}, \qquad i \in \{1,\ldots,C\},\ j \in \{1,\ldots,N\} \quad (1)$$

$$\sum_{i=1}^{N} x_{ij} = 1, \qquad j \in \{1,\ldots,N\} \quad (2)$$

$$\sum_{j=1}^{N} w_j x_{ij} \leq B \qquad i \in \{1,\ldots,C\} \quad (3)$$

Constraint (1) is self-explanatory. Constraint (2) indicates that a given aggregate cannot be assigned to more than one replica set member. Constraint (3) (disjointness constraint) *substitutes the classic capacity constraint of the knapsack problem* and indicates that a given member cannot hold two non-disjoint aggregates. When no confusion is possible, the terms item / aggregate, knapsack / RS member, and profit / interestingness are used interchangeably in the sequel.

To solve the aggregates selection problem, we opt for a depth-first branch-and-bound approach. The pseudo-code of our algorithm is shown in Fig. 4. In this algorithm, the decision tree's successive levels are built by selecting a branching aggregate and assigning it to each knapsack in turn. The branching aggregate is first assigned to a dummy knapsack (lines 1–2), implying its exclusion from the current solution. The aggregate is then assigned to each knapsack that satisfies constraint (2) (lines 3–8). If constraint (2) is satisfied (*i.e.* the conjunction of the knapsack bitmap and the item bitmap is 0), B&B is called recursively to continue exploring the corresponding sub-tree. In the other case, the branch is ignored. This is illustrated in the example of Fig. 5, where knapsack 0 is the dummy knapsack and aggregates a_1 and a_2 are not disjoint. The branches that assign a_1 and a_2 to the same "non-dummy" knapsack are discarded from further consideration (node 1.1 and node 2.2).

In case we have explored all items in a given path, we check if we have a greater

profit than before and update the optimal solution (lines 9-11). Once the set of aggregates assigned to each member is determined, we have to check the recoverability constraint. Indeed, if the disjunction of the bitmaps of aggregates assigned to a knapsack K_i is a bitmap different than B, this means that one or more modeled entity is not represented in K_i. In such a case, an atomic aggregate for each of those entities is added to the member.

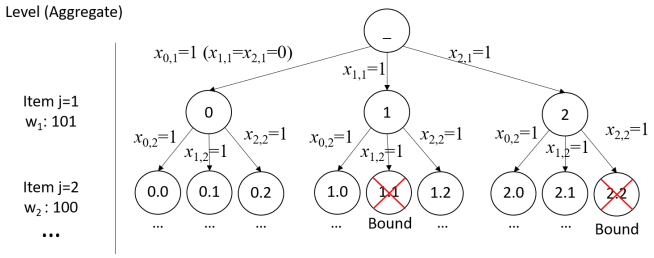


Fig. 5. Item 1 and item 2 are not disjoint. The branches that assign them to the same knapsack are discarded (Branch&Bound).

4. RELATED WORK

4.1 Trojan Data Layouts

[8] proposes Trojan Layouts, a data layout inspired by PAX (Partition Attributes Across) and intended to improve data access times in Hadoop Distributed File System (HDFS). Given a relation R with arity n, PAX partitions each block into n mini-blocks. The ith mini-block stores all the values of the ith attribute of R. Trojan Layouts split an HDFS block into $m \leq n$ mini-blocks and store in each mini-block the values of k ($1 \leq k \leq n$) attributes (*i.e.* vertical partitioning inside each chunk). Trojan Layouts provide a high degree of spatial locality when the values of the k grouped attributes are sequentially accessed and avoid to read $(n-k)$ irrelevant attributes for a given query. To better handle a mix of queries with different access patterns, [8] also proposes to group attributes differently in each HDFS block replica according to the query workload.

Trojan Layouts reorganize a modeled entity's attributes at the block level, while distorted replicas reorganize entities at the database level. Trojan Layouts are then only useful for queries touching a limited subset of an entity's attributes, whereas distorted replicas are useful for queries touching more than one entity.

4.2 Secondary Indexes and the Divergent Physical Design

Numerous techniques have been proposed to build secondary indexes on NoSQL databases. In [17] the authors propose a novel tuning paradigm for replicated databases, called *divergent designs*. Given a replicated database, a divergent design indexes the same data differently in each replica, and hence, specializes replicas for different subsets of the workload [17]. With this design, each query is routed to the replica that can evaluate it most efficiently. The idea of divergent design was further developed in [18], where the authors proposed RITA, an index-tuning advisor for replicated databases. RITA allows to: (i) generate fault-tolerant divergent designs; and (ii) spread the load evenly over replicas.

With secondary indexes, some of the latency would be hidden, but performance would only be sub-optimal since we're still "working against the aggregate structure" as aptly stated in [1] (a further discussion on secondary indexes is given in Section 6).

4.3 Schema Modeling

Despite the wide diffusion of document-oriented DBs, little work has been devoted to their modeling [15]. The work of [9] introduces an aggregates-based logical data model, called NoAM (*NoSQL Abstract Model*), and demonstrates how data modeled in NoAM can be implemented in different NoSQL types. The NoAM modeling approach consists of four steps: (i) aggregate design: the classes of aggregated objects needed for an application are identified (conducted by use cases and performance requirements); (ii) aggregate partitioning: aggregates are partitioned into smaller data elements; (iii) high-level NoSQL DB design: aggregates are mapped to the NoAM model according to the identified partitions; and (iv) implementation: the NoAM schema is converted to the schema of the target NoSQL DB type. Although [9] admits that "*aggregate design is mainly driven by data access operations*", it does not provide a practical workload-aware, cost-based approach for identifying aggregates.

The work of [4] explores different schema design tactics and provides general guidelines for modeling document-oriented DBs. Notably, [4] advocates a workload-aware design consisting of two phases: (i) leveraging common heuristics to generate a finite number of candidate schemas (*i.e. candidate generation*); and (ii) ranking these candidate schemas using cost functions (*i.e. candidate ranking*). [4] states that there are two approaches to candidate generation: *top-down* and *bottom-up*. The bottom-up approach starts with a normalized schema and optimizes each query by adding denormalized structures. The top-down approach starts with a set of globally optimal aggregates that answers each query with a single look-up. Our work follows the broad lines of [4] and can be seen as an implementation of the top-down approach. In our work, however, we go further than [4] (and [9]) and propose a set of practical algorithms and cost functions for the identification, evaluation, and selection of aggregates.

4.4 Database Migration from RDBMS to Document-Oriented NoSQL

As mentioned previously, join operations are rarely supported in document-stores and are often processed at the application layer, in a much more expensive way than RDBMS. Unsurprisingly, most of the existing work on database migration from RDBMS to document-oriented NoSQL [14, 15] propose variants of the denormalized form and try to best balance the trade-offs between the normalized form and the denormalized form. We distinguish two types of denormalization: *table-level denormalization* and *column-level denormalization*. For table-level denormalization, [14] proposes a Breadth-First Search algorithm to find a path from a root table (*i.e.* root entity) to other related tables. This path is used for creating a document template for the root table (by recursively embedding, in a Breadth-First Search fashion, related tables). To reduce the number of embedded entities, the approach of [14], referred to as BFS in the sequel, exploits each link between entities not more than once (an example is given in Fig. 8 (b), where the entity *Region* is embedded only once). As noticed in [19], BFS adds too many weakly related tables in the root document.

The most recent work on denormalization [15] proposed CLDA (*Column-Level Denormalization with Atomicity*), a column-level based denormalization. Rather than embedding entire entities inside aggregates, CLDA preserves all original entities in separate aggregates and duplicates only those columns accessed in non-primary-foreign-key-join predicates (*e.g.* filter predicates). By doing so, CLDA aims at avoiding join operations while minimizing the read-overhead. CLDA can be considered as an implementation of the bottom-up approach discussed in [4]. An important downside of CLDA is that it introduces data redundancy, which is often a source of inconsistency. In contrast with our approach, CLDA is not cost-driven, *i.e.* does not take into account neither the relative importance of queries nor their costs. Furthermore, CLDA systematically duplicates columns without considering the cases where the gain achieved through duplication is lower than the loss caused by the read-overhead. An experimental evaluation of BFS [14], CLDA [15], and distorted replicas is given in Section 5.

5. EXPERIMENTAL EVALUATION

Distorted replicas were implemented on top of MongoDB release 3.6. Experiments were performed on dedicated dual-core i5-3230M systems, running Ubuntu 16.04.1 LTS. Each core offers a base speed of 2.6 GHz and the two cores can handle up to four simultaneous threads. These computers feature 16 GB main memory (DDR3-1600MHz), 128 kB L1 cache, 512 kB L2 cache, and 3 MB L3 cache. The hard disk is a Serial-ATA/600 having a rotational speed of 7200 rpm. MongoDB was run using its default settings and no special tuning was done. All queries were implemented using the MongoDB Aggregation Pipeline. For each query, we report the average execution time of three consecutive runs. We ran our experiments with two main objectives in mind: (i) to show that distorted replicas allow improving data access performance significantly (Subsection 5.1), and (ii) to evaluate the effectiveness of our aggregates selection algorithm (Subsection 5.2).

5.1 Distorted Replicas Effectiveness

We compared distorted replicas with two state-of-the-art approaches: BFS [14], a table-level denormalization method, and CLDA [15], a column-level denormalization method. BFS, CLDA, and distorted replicas were evaluated using two complementary real-world datasets: DBLP [3], a relatively simple workload, and TPC-H [11], a more complex workload. In this paper, we are focusing on reorganizing data within the same replica set and are less concerned with sharding[1]. We then assume that we are given a MongoDB cluster consisting of one replica set or a MongoDB cluster consisting of several shards, each deployed as a replica set. Our aim is to replicate the database hosted at a replica set in different ways and to evaluate the gains in execution time.

5.1.1 DBLP

The DBLP Computer Science Bibliography dataset [3] contains bibliographic information on scientific publications. All the DBLP records are distributed in one big XML file. Each record is associated with a set of fields representing bibliographic data relevant

[1]The study of the effect of data movements between shards (between replica-sets) is part of our ongoing work.

with respect to its type and has an *id* field that uniquely identifies it. We developed a DBLP parser in Java following the recommendations of [3]. Currently, our parser only extracts "article" and "inproceeding" records. The extracted records were inserted in the same collection. The resulting MongoDB collection contains $\approx$ 3.1 million publication documents, embedding $\approx$ 1.6 million distinct authors, and $\approx$ 9.5 thousand distinct journals/conferences. The average document size is 538 bytes, and the total collection size is $\approx$ 1.5 GB. The considered workload is intentionally basic, so that the fundamental properties of CLDA, BFS, and distorted replicas can be better illustrated and highlighted. The workload consists of the following queries.

q_1: *"Find the authors of a given publication"*. In the parsed DBLP dataset, the average number of authors per publication is $\approx$ 2.85. We randomly selected 3 publications co-written by 3 authors and reported the average execution time.

q_2: *"Find the publication titles of a given author"*. The average number of publications per author is $\approx$ 5.46. We randomly selected 3 authors with 6 publications each and measured the average execution time. As authors may have variations in their first names (*e.g.* "Mike Stonebraker", "Michael Stonebraker", *etc.*), we used a regular expression to find out publications (*e.g. author._id:* {*/Stonebraker$/*}).

q_3: *"Find the number of publications per year"*.

The considered BFS aggregate, depicted in Fig. 6 (a), is rooted at *Publication* and embeds *Author* and *Journal*. As shown in Fig. 7, BFS is slightly slower than CLDA and distorted replicas for q_1. This is due to the fact that BFS embeds *Journal* within the aggregate rooted at *Publication*, whereas CLDA and distorted replicas do not. Fig. 7 also shows that BFS outperforms CLDA for q_3 but turns out to be very slow when used to answer q_2. The poor performance of BFS for q_2 is caused by the necessity to visit each *Publication* document and, for each document, to iterate over its *Authors* array (analogously, if the BFS aggregate were rooted at *Author*, it would be slightly slower than CLDA and distorted replicas for q_2, but very slow for q_1 and q_3).

CLDA is optimal for q_1 and q_2. However, in the absence of a query involving *Publication* and containing a filter predicate on *Journal.year*, the relationship between *Publication* and *Journal* is not denormalized. Answering q_3 using CLDA requires therefore a costly join operation (Fig. 7).

Fig. 6 (b) illustrates the set of aggregates selected by our approach (for $C = 3$). As shown in Fig. 7, distorted replicas are optimal for q_1 and q_3 and only slightly slower than CLDA for q_2 (as distorted replicas embed the entire *Publication* entity within the aggregate rooted at *Author*, whereas CLDA only embeds *Publication._id* and *Publication.title*). Overall, distorted replicas are, respectively, $\approx$ 2.8 and $\approx$ 1.28 faster than BFS and CLDA for the considered workload (an improvement factor of 2 for a query q or a workload W means that q/W is executed 2 times faster).

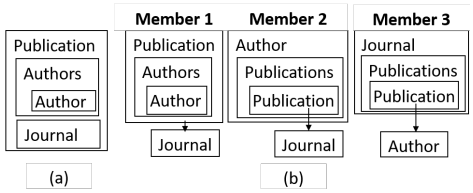


Fig. 6. DBLP dataset; (a) BFS DB; (b) Distorted replicas ($C = 3$).

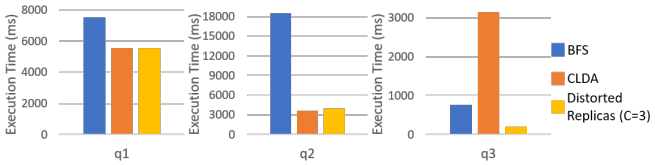


Fig. 7. Query performance (DBLP).

5.1.2 TPC-H

We generated TPC-H data using the TPC-H DBGEN data generator tool with a scale factor of 10 and developed a Java loader module to import TPC-H data in MongoDB.

We considered 10 representative TPC-H queries: q_1, q_3, q_5, q_6, q_{10}, q_{11}, q_{14}, q_{15}, q_{17}, q_{19}. The outputs of our algorithm steps are synthesized in Table 1. The BFS approach could produce different denormalized schemas, depending on the order of edge visits [14, 15]. We adopted the schema described in [15] and illustrated in Fig. 8 (b). As shown in Fig. 8 (b), the BFS DB is very close to a fully denormalized DB having the same root. The only difference is that *Region* is embedded only once.

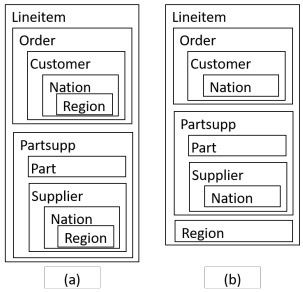


Fig. 8. TPC-H dataset; (a) Fully denormalized DB; (b) BFS DB.

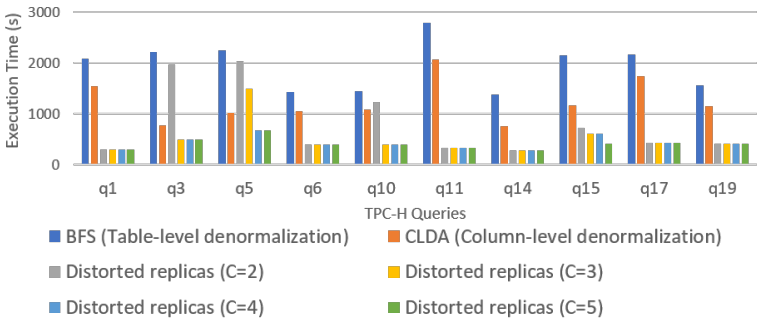


Fig. 9. Query performance (TPC-H).

Fig. 9 depicts the performance achieved by BFS, CLDA and distorted replicas, in terms of query execution time. Fig. 10 reports the improvements achieved by distorted replicas over BFS and CLDA as a function of the replication factor C. As shown in Fig. 9 and Fig. 10, BFS is by far outperformed by CLDA and distorted replicas (resp., ≈ 1.51 and up to ≈ 4.97 times faster). To understand the behavior of distorted replicas and CLDA, let's consider the atomic aggregate $\{Lineitem, (Lineitem)\}$, which is optimal for queries q_1 and q_6 and contributes to optimizing q_{15} and q_{17}. The atomic aggregate $\{Lineitem, (Lineitem)\}$ is selected by our knapsack algorithm $\forall\, C \geq 2$. Due to queries

such as q_5, q_{14}, and q_{19}, the relationships between *Lineitem*, *Supplier*, *Part*, and *Nation* are partly denormalized in CLDA (*i.e.* some columns from *Supplier*, *Part*, and *Nation* are embedded within *Lineitem*). The embedded columns are useless for queries q_1, q_6, q_{15} and q_{17}, and introduce a substantial read-overhead. Furthermore, CLDA nests *Lineitem* into *Order* as an array of sub-documents to better support atomicity [15]. Answering q_1, q_6, q_{15}, and q_{17} using CLDA, requires therefore an expensive *unwind* operation (to flatten the array of sub-documents).

As expected and shown in Fig. 9, distorted replicas outperform CLDA for these queries. Fig. 9 also shows that when C is low, CLDA performs better than distorted replicas for some queries (*e.g.* q_5 and q_{15}). This is due to the disjointness constraint, which does not allow to select all optimal aggregates. As shown in Fig. 9, when C is increased, more optimal aggregates are selected and more queries are optimized.

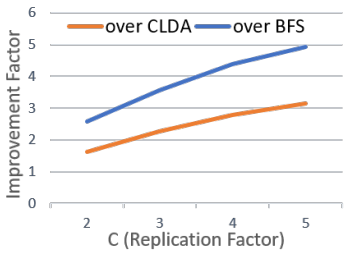


Fig. 10. Improvement factor as a function of the Replication Factor.

5.2 Selection Algorithm Performance

Now we focus on the effectiveness of our algorithm of aggregates selection.

5.2.1 Number of Iterations.

First of all, we show the effect of adding the disjointness constraint to our knapsack formulation. Recall that the "disjointness constraint" substitutes the classic capacity constraint of the knapsack problem and is used to prune the search space. Table 2 compares the number of iterations with and without the disjointness constraint as a function of the replication factor (*i.e.* number of knapsacks). The number of iterations without the disjointness constraint corresponds to the total number of nodes in the decision tree. As shown in Table 2, the disjointness constraint substantially reduces the number of iterations in our algorithm: the fraction of visited nodes for $C = 5$ is $\approx$ 1.17803E-07.

Table 1. Aggregates selection algorithm applied to TPC-H queries.

# of queries	# of relevant aggregates	# of merged pairs	# of selected aggregates (C=2)	# of selected aggregates (C=3)	# of selected aggregates (C=4)	# of selected aggregates (C=5)
10	18	3	3	4	5	6

Table 2. Knapsack iterations as a function of the replication factor C (N=21 items).

	C=2	*C=3*	*C=4*	*C=5*
With Disjointness Constraint	47 600	2 252 212	84 579 458	2 584 233 662
Without Disjointness Constraint	10 460 353 203	4.39805E+12	4.76837E+14	2.1937E+16

Table 3. Execution time as a function of the replication factor C (N=21 items).

C=2	*C=3*	*C=4*	*C=5*
794 ms	8 s	302 s	172 m

5.2.2 Execution time

Table 3 shows the time taken to create aggregate groups, as a function of the replication factor C. The execution time of our algorithm is highly dominated by the resolution of the MKP problem using the branch-and-bound algorithm. For 21 interesting aggregates and 5 replicas, the time taken to group aggregates is around 172 minutes. This is acceptable, given that grouping is an offline process. Recall that, as suggested in Subsection 3.3, the execution time can be reduced by discarding aggregates whose interestingness is below a predefined threshold.

6. DISCUSSION

Like materialized views and secondary indexes, distorted replicas are intended to improve query performance for workloads of common and repeated query patterns. In this work, we showed that distorted replicas substantially reduce query execution time. A salient feature of our work is that it maps the problem of aggregates packing to a 0-1 Multiple Knapsack problem and solves it using a branch and bound technique.

Typically, database systems gather statistics on search queries and provide tools for diagnosing database performance. In the same way as materialized views and secondary indexes, database admins can generate distorted replicas (and eventually delete less useful ones) when they witness slow execution times due to new query patterns. Distorted replicas can also be generated when a database is migrated from RDBMS to NoSQL or at design time (in the latter case, the DB designer will have to estimate relationship cardinalities and document sizes). As materialized views, distorted replicas restructure documents to provide new ways for exploring data. The main difference is that distorted replicas take advantage of the already-existing replication to generate restructured data without any additional refresh cost, while materialized views introduce a significant write overhead. It is worth noticing that if materialized views were enabled to regenerate base data in a two-way replication scenario, they would be an interesting tool to implement distorted replicas (such a scenario is permitted in some RDBMSs such as Oracle).

Compared to distorted replicas, secondary indexes only allow sub-optimal performance. Consider for example the DBLP dataset where data is organized by *Publication* and a query such as "Find Stonebraker's publications". A secondary index on author identifiers allows only retrieving the subset of documents with an author named Stonebraker. In contrast, with a distorted replica where data is organized by *Author*, only one document embedding all Stonebraker's publications would be retrieved. Secondary indexes are also not helpful for map/reduce jobs and queries requiring a full scan or involving regular expressions. With the DBLP dataset and the MongoDB settings of Section 5, a query looking for the publications of any author having "Stonebraker" as last name (regex: *author._id:* {*/Stonebraker$/*}), takes on average 49980 ms with a secondary index, 18505 ms without an index and only 3849 ms if data is organized by *Author*.

7. CONCLUSION

Aggregates are useful in that they pack into one document, data that is expected to be accessed together. Aggregates are essential to data processing at the level of large-scale clusters, but severely limit the ways data can be efficiently explored and processed. This paper deepened the idea of distorted replicas, a replication scheme that restructures replicated data in different ways to improve data access times. In particular, we showed how to: (i) identify relevant aggregates; (ii) assign interestingness values to relevant aggregates; (iii) merge pairs of interesting aggregates; and (iv) pack interesting aggregates in such a way that the total interestingness of a replica set is maximized. We also implemented our ideas on top of MongoDB and evaluated distorted replicas using two real-world datasets. Experimental results show that distorted replicas substantially reduce query execution time: up to tens of times faster than state-of-art methods.

As a part of our future work, we intend to define a cost model to help query optimizers determine the replica to which a query should be directed. Another interesting point to consider is a closer study of dynamic migration of data between shards and its impact on distorted replicas and on load balancing.

ACKNOWLEDGMENT

We would like to thank Pr. Ouajdi Korbaa for many helpful discussions and reviews that improved this paper.

REFERENCES

1. P. J. Sadalage and M. Fowler, *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence*, Addison-Wesley Professional, NJ, 2012.
2. A. Corbellini, C. Mateos, A. Zunino, D. Godoy, and S. Schiaffino, "Persisting big-data: The nosql landscape," *Information Systems*, Vol. 63, 2017, pp. 1-23.
3. M. Ley, "DBLP – some lessons learned," in *Proceedings of VLDB Endowment*, Vol. 2, 2009, pp. 1493-1500.
4. V. Reniers, D. V. Landuyt, A. Rafique, and W. Joosen, "Schema design support for semi-structured data: Finding the sweet spot between NF and De-NF," in *Proceedings of IEEE International Conference on Big Data*, 2017, pp. 2921-2930.
5. C. de Lima and R. dos Santos Mello, "A workload-driven logical design approach for nosql document databases," in *Proceedings of the 17th International Conference on Information Integration and Web-Based Apps & Services*, 2015, pp. 1-10.
6. K. Jouini, "Distorted replicas: Intelligent replication schemes to boost I/O throughput in document-stores," in *Proceedings of IEEE/ACS 14th International Conference on Computer Systems and Applications*, 2017, pp. 25-32.
7. R. Ramamurthy, D. J. DeWitt, and Q. Su, "A case for fractured mirrors," *The VLDB Journal*, Vol. 12, 2003, pp. 89-101.
8. A. Jindal, J. Quiané-Ruiz, and J. Dittrich, "Trojan data layouts: right shoes for a running elephant," in *Proceedings of ACM Symposium on Cloud Computing*, 2011, p. 21.

9. P. Atzeni, F. Bugiotti, L. Cabibbo, and R. Torlone, "Data modeling in the NoSQL world," *Computer Standards & Interfaces*, Vol. 6, 2020, pp. 103-149.
10. V. Varga, C. Sacarea, and A. É. Molnár, "Conceptual graphs based modeling of semi-structured data," in *Proceedings of the 23rd International Conference on Conceptual Structures*, LNCS 10872, 2018, pp. 167-175.
11. "The TPC-H benchmark," http://www.tpc.org/tpch, 2020.
12. K. Jouini, G. Jomier, and P. Kabore, "Read-optimized, cache-conscious, page layouts for temporal relational data," in *Proceedings of the 19th International Conference on Database and Expert Systems Applications*, LNCS 5181, 2008, pp. 581-595.
13. "MongoDB," http://www.mongodb.com/guides/, 2019.
14. G. Karnitis and G. Arnicans, "Migration of relational database to document-oriented database: Structure denormalization and data transformation," in *Proceedings of the 7th International Conference on Computational Intelligence, Communication Systems and Networks*, 2015, pp. 113-118.
15. J. Yoo, K. Lee, and Y. Jeon, "Migration from RDBMS to NoSQL using column-level denormalization and atomic aggregates," *Journal of Information Science and Engineering*, Vol. 34, 2018, pp. 243-259.
16. S. Chaudhuri and V. R. Narasayya, "Self-tuning database systems: A decade of progress," in *Proceedings of the 33rd International Conference on Very Large Data Bases*, 2007, pp. 3-14.
17. M. P. Consens, K. Ioannidou, J. LeFevre, and N. Polyzotis, "Divergent physical design tuning for replicated databases," in *Proceedings of ACM SIGMOD International Conference on Management of Data*, 2012, pp. 49-60.
18. Q. T. Tran, I. Jimenez, R. Wang, N. Polyzotis, and A. Ailamaki, "RITA: an index-tuning advisor for replicated databases," in *Proceedings of the 27th International Conference on Scientific and Statistical Database Management*, 2015, pp. 22:1-22:12.
19. B. Namdeo and U. Suman, "Performance analysis of schema design approaches for migration from RDBMS to NoSQL databases," in *Advances in Data and Information Sciences*, 2020, pp. 413-424.

Khaled Jouini received the Ph.D. degree in Computer Science from Paris-Dauphine University, France. He was a research staff member at Telecom ParisTech, France. Since 2011, he has been with Sousse University, Tunisia, where he is currently an Associate Professor. His research interests include non-volatile memory, database systems, and large-scale data management and mining.

.

Publications dans des conférences classées

. .

B.1 Enhancing LULC Classification with Attention-Based Fusion of Handcrafted and Deep Features

Vian Abdulmajeed Ahmad, Khaled Jouini & Ouajdi Korbaa

17^{th} International Conference on Agents and Artificial Intelligence (ICAART). Feb 2025. (to appear)

CORE Rank B

Gmail Khaled Jouini <j.khaled@gmail.com>

Fwd: ICAART 2025 - Authors Notification

8 messages

---------- Forwarded message ---------
من: **ICAART Secretariat** <icaart.secretariat@insticc.org>
Date: الأربعاء، ٤ ديسمبر ٢٠٢٤, ٦:٢٥ م
Subject: ICAART 2025 - Authors Notification
To: <vian.ahmad85@gmail.com>

Dear Eng. Vian Ahmed,

We are happy to inform you that the regular paper you have submitted to ICAART, with number 51, entitled "Enhancing LULC Classification with Attention-Based Fusion of Handcrafted and Deep Features", has been accepted as a Full Paper.
Papers accepted as Full Paper are assigned a 12-page limit in the conference proceedings.

All reviews performed by the program committee are now available at the PRIMORIS Author's Home https://www.insticc.org/Primoris/. Please login and then click on Author's home / Paper Reviews, to access the reviews.
The e-mail associated with your account is also your username: vian.ahmad85@gmail.com

It is very important that you try to follow the suggestions indicated in the reviews during the preparation of the camera-ready manuscript.
Furthermore, it is EXTREMELY important that you follow the camera-ready paper format and preparation guidelines for the proceedings, which are available at the ICAART web site https://icaart.scitevents.org/Guidelines.aspx.

Any non-conformance with the specified format may force the proceedings editing team to return the paper to you for re-formatting, and in case of repeated problems it may prevent your paper from being published altogether.

Concerning the presentation of your Full Paper, it was recommended that it should be presented in the format of a 25 min. oral presentation (including discussion). Please prepare it according to the instructions available at the conference website.

Please note that the publication of any paper in the conference proceedings requires that:
- we receive the camera ready version of your paper, via Primoris, until December 20, 2024;
- after submitting the camera ready you need to approve the copyright document - an icon named "Copyright" will appear at your "Author's Area";
- one of the authors must be registered as a speaker for this paper before December 20, 2024.

You can only complete your registration after you submit your camera ready; payment can be made during registration, or afterwards in some cases, using:
- Credit Card (directly on our system);
- PayPal (using a credit card - a PayPal account is necessary. Creating one may take up to 5 working days due to the verification process);
- Bank transfer (the transfer should be done after the online registration is complete and it's only valid after the bank transfer document is received);
- Bon de commande/Purchase order (the document should be sent to the secretariat immediately after the online registration is complete).

A detailed explanation regarding each option is available during the registration process. Any queries regarding the registration or invitation letters must be sent to the following email address: registrations@insticc.org.

Furthermore, in your author's area you have a button called Rebuttal. The rebuttal offers authors an opportunity to

Gmail **Khaled Jouini <j.khaled@gmail.com>**

Fwd: Registration to ICAART 2025

1 message

---------- Forwarded message ---------
من: **ICAART 2025 Secretariat** <icaart.secretariat@insticc.org>
Date: الأربعاء، ١١ ديسمبر ٢٠٢٤, ١١:٢٥ م
Subject: Registration to ICAART 2025
To: <vian.ahmad85@gmail.com>

Dear Vian Ahmed,

Thank you for your registration to ICAART 2025.

Please find below the data pertinent to your registration (ref. 74721)

--- Registration Items ---
Conference Early Registration: 565€
- 17th International Conference on Agents and Artificial Intelligence (ICAART 2025)
Paper # 51: Enhancing LULC Classification with Attention-Based Fusion of Handcrafted and Deep Features: included in registration
INSTICC 2025 Membership: free

Registration Amount: 565.00€ (EUR)

--- Invoice/Receipt Information ---
VAT number:
Addressed to: Vian Ahmed
Address: Iraq, Kirkuk
Kirkuk
00964
Iraq

--- Payment ---
Method: Credit Card

Please don't hesitate to contact me with any questions you may have.

Best regards,
ICAART 2025 Secretariat

ICORE Conference Portal

Back to search

International Conference on Agents and Artificial Intelligence

Acronym: ICAART

DBLP Source: https://dblp.uni-trier.de/db/conf/icaart

Source: CORE2023

Rank: B

Field Of Research: 4602 - Artificial intelligence (h-index) (citation)

Source: CORE2021

Rank: B

Field Of Research: 4602 - Artificial intelligence

Request for rank B accepted (Data 1) (Decision)

Source: CORE2020

Rank: C

Field Of Research: 4602 - Artificial intelligence

Source: CORE2018

Rank: C

Field Of Research: 0801 - Artificial Intelligence and Image Processing†

Source: CORE2017

Rank: C

Field Of Research: 0801 - Artificial Intelligence and Image Processing†

Source: CORE2014

Rank: C

Enhancing LULC Classification with Attention-Based Fusion of Handcrafted and Deep Features

Vian Abdulmajeed Ahmed[1][a], Khaled Jouini[1][b] and Ouajdi Korbaa[1][c]
[1]*MARS Research Lab, LR17ES05, ISITCom, University of Sousse, Sousse, Tunisia*
vian.ahmad85@gmail.com, j.khaled@gmail.com, ouajdi.korbaa@centraliens-lille.org

Keywords: Land Use and Land Cover Classification, Attention-based Fusion, Early and Late fusion, Multi-modal Learning.

Abstract: Satellite imagery provides a unique perspective of the Earth's surface, pivotal for applications like environmental monitoring and urban planning. Despite significant advancements, analyzing satellite imagery remains challenging due to complex and variable land cover patterns. Traditional handcrafted descriptors like Scale-Invariant Feature Transform (SIFT) excel at capturing local features but often fail to capture the global context. Conversely, Convolutional Neural Networks (CNNs) excel at capturing rich contextual information but may miss crucial local features due to limitations in capturing small and subtle spatial arrangements. Most existing Land Use and Land Cover (LULC) classification approaches heavily rely on fine-tuning large pretrained models. While this remains a powerful tool, this paper explores alternative strategies by leveraging the complementary strengths of handcrafted and CNN-learned features. Specifically, we investigate and compare three fusion strategies: (*i*) early fusion, where handcrafted and CNN-learned features are merged at the input level; (*ii*) late fusion, where attention mechanisms dynamically integrate salient features from both CNN and SIFT modalities; and (*iii*) mid-level fusion, where attention is used to generate two feature maps: one prioritizing global context and another, weighted by SIFT features, emphasizing local details. Experiments on the real-world EuroSAT dataset demonstrate that these fusion approaches exhibit varying levels of effectiveness and that a well-chosen fusion strategy not only substantially outperforms the underlying methods used separately but also offers an interesting alternative to solely relying on fine-tuning pre-trained large models.

1 INTRODUCTION

Satellite imagery underpins critical applications like land cover mapping, environmental monitoring, disaster response, and urban planning (Ahmed et al., 2024). At the heart of these applications lies *Land Use and Land Cover* (LULC) classification, which involves assigning predefined semantic classes to remote sensing images. Effective LULC classification requires the capability to discern complex spatial patterns while maintaining robustness against variations in scale, atmospheric conditions, and noise (Xia and Liu, 2019). Early methods in LULC classification relied heavily on handcrafted descriptors like *Scale-Invariant Feature Transform* (SIFT) (Lowe, 2004) and similar approaches, which excel at capturing distinctive local features such as edges and textured regions. These methods, however, often struggle to capture the complex spatial and contextual information present in remote sensing images due to their local nature (Cheng et al., 2019). The advent of deep learning and its models trained on large datasets has revolutionized image classification, achieving accuracy levels far beyond traditional methods. The impressive performance of these models, coupled with their data-intensive nature, has directed much of the current work on LULC classification towards *transfer learning* (Dewangkoro and Arymurthy, 2021; Helber et al., 2019; Wang et al., 2024; Neumann et al., 2020), which involves fine-tuning pre-trained large models on remote sensing datasets.

The *convolutional layers* of deep models operate by applying learned filters (small grids of weights) that slide across the image. As these filters move, their weights are multiplied element-wise with pixel values, and the results are summed to create a *feature map*. The stacking of convolutional layers enables deep models to learn increasingly complex patterns: early layers typically capture basic elements

[a] https://orcid.org/0009-0002-5924-6139
[b] https://orcid.org/0000-0001-5049-4238
[c] https://orcid.org/0000-0003-4462-1805

like edges and corners, while subsequent layers interpret these elements to recognize objects. This hierarchical learning process allows Convolutional Neural Networks (CNNs) to excel at capturing global contexts and spatial relationships. Despite these capabilities, CNNs can struggle to preserve very fine-grained details like small textures or subtle variations in color due to the pooling operations that often follow convolutional layers.

In this work, we aim to investigate and quantify the potential benefits of synergizing the complementary strengths of handcrafted SIFT descriptors and features learned from CNNs. Fusing these features is intended to leverage both the local detailed cues provided by SIFT and the global context captured by CNNs. Specifically, we investigate three fusion strategies: a straightforward early fusion approach, and novel late and mid-level fusion approaches integrating *attention mechanisms*. Attention mechanisms enable neural networks to prioritize informative input elements by assigning them weights that reflect their relative importance. The late fusion approach uses attention to dynamically weigh and integrate salient features from both the CNN and SIFT modalities before making final classification decisions.The mid-level fusion approach generates two distinct feature maps: one prioritizing global context and another locally-attended feature map weighted according to SIFT features, emphasizing local details. Our experimental study on the real-world EuroSAT dataset reveals that the different fusion approaches vary in effectiveness. Our study also suggests, that while the prevalent fine-tuning of pre-trained models remains a powerful tool for LULC classification, alternatives such as integrating handcrafted and CNN-learned features warrant exploration.

The remainder of this paper is organized as follows. Section 2 provides a concise overview of previous research. Section 3 introduces our features fusion approaches. Section 4 presents a comparative experimental analysis. Finally, Section 5 concludes the paper.

2 RELATED WORK

The EuroSAT dataset (Helber et al., 2018) is a widely recognized and extensively used dataset for LULC classification. It includes 27,000 geotagged image patches, each covering an area of 64x64 meters with a spatial resolution of 10 meters. The dataset comprises ten distinct classes, with each class including 2,000 to 3,000 images. As illustrated in Figure 1, these classes represent a diverse range of land use and land cover types. For the sake of conciseness and due to lack of space, we mainly focus in the sequel on approaches presenting similarities with our work or that use EuroSAT. Existing remote sensing image classification approaches and studies can be broadly classified into two families: Machine Learning (ML)-based and Deep Learning (DL)-based methods.

The study by Chen & Tian (Chen and Tian, 2015), and Thakur & Panse (Thakur and Panse, 2022) are representative of ML-based approaches. (Chen and Tian, 2015) introduced the Pyramid of Spatial Relations (PSR) model, designed to incorporate both relative and complete spatial information into the BoVW (*i.e.* Bag of Visual Words) framework. Experiments conducted on a high-resolution remote sensing image revealed that the PSR model achieves an average classification accuracy of 89.1%. In (Thakur and Panse, 2022), the performance of four machine learning algorithms was evaluated on the EuroSAT dataset: Decision Tree (DT), K-Nearest Neighbour (KNN), Support Vector Machine (SVM), and Random Forest (RF). The study revealed distinct performance levels among the algorithms: RF achieved the highest overall accuracy of 56.70%, significantly outperforming DT and KNN.

The studies (Temenos et al., 2023), (Dewangkoro and Arymurthy, 2021), (Helber et al., 2019), (Wang et al., 2024) and (Neumann et al., 2020) are representative of DL-based approaches. In (Temenos et al., 2023), the authors introduce an interpretable DL framework for LULC classification using SHapley Additive exPlanations (SHAPs). They employ a compact CNN model for image classification, followed by feeding the results to a SHAP deep explainer, achieving an overall accuracy of 94.72% on EuroSAT. The approach in (Dewangkoro and Arymurthy, 2021) utilizes different CNN architectures for feature extraction, including VGG19, ResNet50, and InceptionV3. These extracted features are then recalibrated using the Channel Squeeze & Spatial Excitation (sSE) block, with Twin SVM (TWSVM) serving as classifier, achieving an accuracy of 94.39% on EuroSAT. In (Helber et al., 2019), various CNN architectures were compared, including a shallow CNN, a ResNet50-based model, and a GoogleNet-based model. The achieved classification accuracies on EuroSAT were 89.03%, 98.57%, and 98.18%, respectively. (Neumann et al., 2020) explored in-domain fine-tuning using five diverse remote sensing datasets and the ResNet50V2 architecture. (Neumann et al., 2020) demonstrated that models fine-tuned on in-domain datasets significantly outperform those pretrained on general purpose datasets like ImageNet. The pretrained ResNet50v2 fine-tuned on in-domain

Figure 1: Sample Images Extracted from the EuroSAT Dataset (Helber et al., 2019)

datasets achieved an overall accuracy of 99.2% on EuroSAT.

While transfer learning typically involves adapting a pre-trained model to improve performance on a related dataset, knowledge transfer involves training a single model on multiple tasks simultaneously and leveraging shared representations and knowledge across these tasks. (Gesmundo and Dean, 2022) used knowledge transfer and employed a multitask learning framework in which the model learns from diverse remote sensing datasets concurrently. The evolutionary "mutant multitask network" (μ2Net), introduced by (Gesmundo and Dean, 2022), enhances model efficiency and quality through effective knowledge transfer mechanisms while addressing common challenges such as catastrophic forgetting and negative transfer. Empirical results demonstrate that μ2Net can achieve competitive performance across various image classification tasks. Specifically, on EuroSAT, μ2Net achieved a high classification accuracy of 99.2%. Knowledge transfer is also used by (Wang et al., 2024), where Vision Transformers (ViT) (Steiner et al., 2021) with Rotatable Variance Scaled Attention (RVSA) are used as part of a Multi-Task Pretraining (MTP) framework. When evaluated on EuroSAT, the MTP-enhanced model achieved a high accuracy of 99.2%.

As outlined in this section, most existing LULC classification approaches typically focus on either classical ML or DL methods. Studies (Tianyu et al., 2018) and (Ahmed et al., 2024) have demonstrated the benefits of combining handcrafted and CNN-learned features on the general-purpose CIFAR dataset and EuroSAT dataset, respectively. However, these studies only explored a straightforward early fusion approach. Our work proposes more advanced attention-based fusion methods that can potentially learn to focus on more discriminative features for improved LULC classification accuracy.

3 SYNERGIZING HANDCRAFTED AND CNN LEARENED FEATURES

As mentioned earlier, SIFT is adept at capturing intricate local details and textures but falls short in interpreting broader scene contexts. In contrast, CNNs excel at understanding contextual information and spatial relationships, yet they may overlook fine-grained details. Integrating these features potentially allows the model mitigating the limitations of each method when used alone.

The remainder of this section explores three distinct fusion strategies: (straightforward) early fusion, and novel late and mid-level fusion with attention mechanisms. Broadly, early fusion directly combines features extracted from different modalities before feeding them into a classifier. Conversely, late fusion extracts features independently using separate models for each modality and then fuses these features before classification. Mid-level fusion partially extracts features from each modality before allowing information exchange between them, enabling them to influence each other's feature learning process.

3.1 Baseline Models and Early Fusion Approach

This section provides an overview of the baseline models and the early fusion approach used in our experiments, establishing a foundation for understanding the more advanced late and mid-level fusion approaches discussed later in this paper.

SIFT identifies keypoints in an image that remain stable under scale, rotation, and illumination changes. Initially, SIFT creates a scale-space representation of the image by convolving it with Gaussian filters at multiple scales. Keypoints are then localized as local extrema (peaks or valleys) in the Difference-of-Gaussian (DoG) images computed across these scales (Lowe, 2004). Keypoints are typically found at corners, edges, or distinct texture patterns (Lowe, 2004) and in areas with significant variations in intensity across different directions. In the context of satellite images, keypoints often correspond to transitions between different land covers. To enhance accuracy, each keypoint's precise position and scale are refined through interpolation to achieve subpixel accuracy. Once keypoints are identified, SIFT computes a descriptor for each of them. This descriptor encapsulates information about the gradients or directional changes in intensity surrounding that keypoint within a localized patch of the image (Lowe, 2004). The standard SIFT descriptor is generated by creating a histogram of gradient orientations within this patch, divided into a 4×4 grid, with each of the 16 cells contributing eight orientation bins, resulting in a 128-element descriptor vector. SIFT identifies potentially hundreds or thousands of keypoints per image. To simplify data representation and ensure compatibility with most machine learning algorithms, all individual keypoint descriptors are typically concatenated into a single row vector (*flattening*).

Figures 2 and 3 illustrate the baseline models employed in our work. The first baseline model is a neural network that takes SIFT descriptors as input. It follows a common architecture with two dense layers for feature processing, using ReLU (Rectified Linear Unit) activation functions to introduce non-linearity. Batch normalization is incorporated to stabilize training, and dropout with L2 regularization are applied to prevent overfitting. The second baseline model is a convolutional neural network (CNN) that takes RGB images as input. It features a standard architecture comprising convolutional layers for feature extraction, pooling layers for downsampling, a flattening layer for feature vector transformation, and fully-connected layers for classification. ReLU activation functions are used throughout the network to introduce non-linearity, and dropout with L2 regularization are applied to prevent overfitting.

Figure 4 illustrates the early fusion model used in our work. Early fusion is a prevalent approach for combining features extracted from different modalities (Ahmed et al., 2024) and consists in combining these features before feeding them into the higher-level layers of a neural network. Despite its simplicity, early fusion can achieve good accuracy because it enables the model to learn a unified representation that leverages information from both RGB images and SIFT descriptors during the training process (Ahmed et al., 2024). As shown in Figure 4, the early fusion model we employed comprises two distinct branches, each processing a different modality. After feature extraction in each branch, the model concatenates them. This fused feature vector is then passed through standard neural network layers that integrate regularization techniques (dropout, L2). The final layer employs a softmax activation function for multi-class classification.

3.2 Late Fusion: Attention-Enhanced Dual Learning (ADL)

Attention mechanisms enable neural network models to selectively focus on informative aspects within the input data. They achieve this by learning a set of weights for different parts of the input, indicating their relative importance. Broadly, attention mechanisms operate by calculating scores (*e.g.*, element-wise multiplication, dot products) reflecting the potential relevance of each element in the input. These scores are learned dynamically based on the context. A softmax function is then applied to the scores, normalizing them into a probability distribution. The resulting weights sum to 1 and represent the relative importance of each element as a probability. Finally, the original input elements are multiplied by their corresponding weights and then summed. This creates a weighted representation of the input, emphasizing informative aspects based on the learned attention weights.

As depicted in Figure 5, in our proposed late fusion model, features are extracted from each modality independently using separate models, and then the outputs of the branches are fused at the very end of the network before classification. To improve feature learning, our late fusion model leverages adequate attention mechanisms in both branches. A *channel-wise attention* is integrated into the CNN of the RGB branch through *Squeeze-and-Excitation* (SE) (Hu et al., 2018) blocks. The SE block dynamically adjusts the importance of each channel within

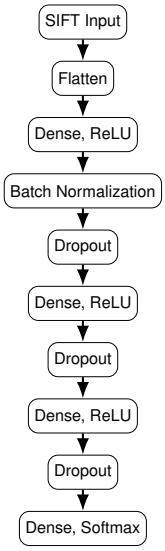


Figure 2: Baseline SIFT-NN Model

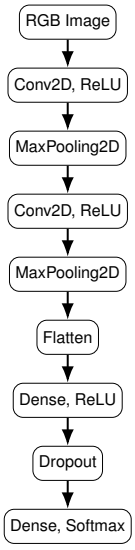


Figure 3: Baseline Shallow CNN Model

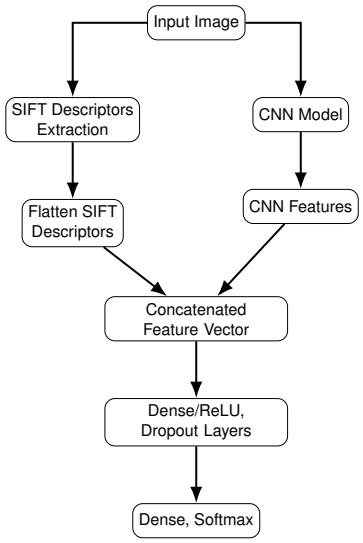


Figure 4: Early Fusion of SIFT and CNN Features

the feature maps. The process involves the following steps (Hu et al., 2018):

1. *Squeeze*: Global average pooling is applied to each feature map, reducing each channel to a single value: $z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_{i,j,c}$, where $x_{i,j,c}$ represents the value at position (i, j) in channel c, and H and W are the height and width of the feature map.
2. *Excitation*: A gating mechanism with a bottleneck structure (two fully connected layers) is applied to capture channel-wise dependencies: $s = \sigma(W_2 \delta(W_1 z))$, where W_1 and W_2 are the weight matrices, δ denotes the ReLU activation function, and σ denotes the sigmoid activation function.
3. *Recalibration*: The original feature map is scaled by the learned channel weights: $\tilde{x}_{i,j,c} = s_c \cdot x_{i,j,c}$

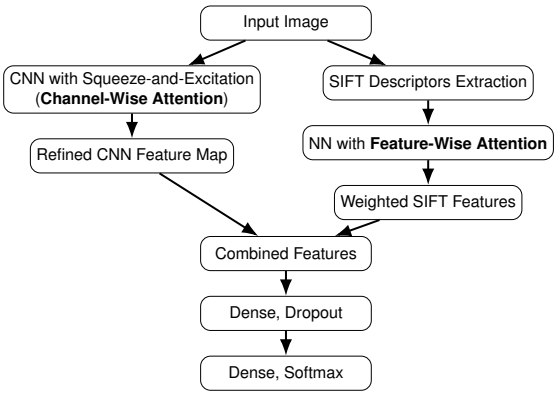


Figure 5: Late Fusion Approach : Attention-Enhanced Dual Learning (ADL)

In the SIFT branch, SIFT descriptors are processed using a neural network integrating a *feature-wise attention layer* that assigns weights to descriptors to emphasize the most informative ones. By assigning higher weights to relevant SIFT descriptors, the attention mechanism emphasizes features that are particularly informative for specific LULC classes. This complements the focus on global spatial relationships learned by the CNN in the RGB branch. Formally, each SIFT descriptor x_i is transformed into query Q_i, key K_i, and value V_i vectors using learned linear transformations:

$$Q_i = W_Q x_i + b_Q, \quad K_i = W_K x_i + b_K, \quad V_i = W_V x_i + b_V$$

where W_Q, W_K, W_V are weight matrices and b_Q, b_K, b_V are bias vectors. These transformations enable the network to effectively compute attention scores, which measure the relevance of each feature within the descriptor relative to others. The attention score for each feature within the SIFT descriptor is computed as the dot product of the query vector with all key vectors: $score_{ij} = Q_i \cdot K_j$. This results in a matrix of attention scores indicating the relevance of each feature with respect to all others. The subsequent softmax normalization of these scores produces attention weights that denote the significance of each descriptor element: $\alpha_{ij} = \frac{\exp(score_{ij})}{\sum_k \exp(score_{ik})}$. Ultimately, a weighted sum of the value vectors, weighted by these attention weights, results in a refined representation of the SIFT descriptors. The final output of the attention mechanism is the weighted sum of the value

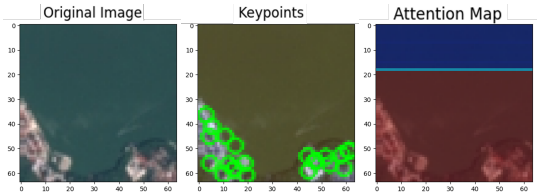


Figure 6: Example Illustrating SIFT-Guided Attention.

vectors, where the weights are the normalized attention scores: $attended\_features_i = \sum_j \alpha_{ij} V_j$.

After the independent processing in each branch, the enriched feature outputs from the RGB branch (with channel-wise attention) and the SIFT branch (with feature-wise attention) are concatenated. The fused features combine the spatial relationships captured by the CNN with the local variations captured by the SIFT descriptors, enriched by their respective attention mechanisms. The concatenated feature vector serves as input to dense layers with regularization techniques (dropout, L2).

3.3 Mid-Level Fusion: Fusion of Local Attended CNN Features and Global CNN Features (LFGF) with Gating Mechanism

Early fusion as well as late fusion combine independent feature representations from CNN and SIFT for classification, considering both global and local features equally important and informative. In this section, we propose a novel mid-level fusion approach with the same aim of leveraging local SIFT cues to help identify potentially informative regions, but with a different rationale.

Instead of concatenating separate SIFT and CNN features, the proposed mid-level approach fuses *CNN global features*, which capture the global context, with *localized, attention-weighted CNN features* that specialize in capturing finer-grained local details. To achieve this, we use a custom *SIFT-guided dynamic* and *adaptive attention* mechanism.

As illustrated in the example of Figure 6 (extracted from our experiments), within this attention mechanism, SIFT descriptors and keypoints act as guides, highlighting potentially informative regions within the image that are likely to hold discriminative power for distinguishing between different land cover types. The attention mechanism subsequently focuses on these highlighted areas, selectively amplifying the detailed features captured by the CNN within those specific patches. Most importantly, this approach integrates, rather than discards, the rich feature set extracted by the CNN from the entire image and fuses it

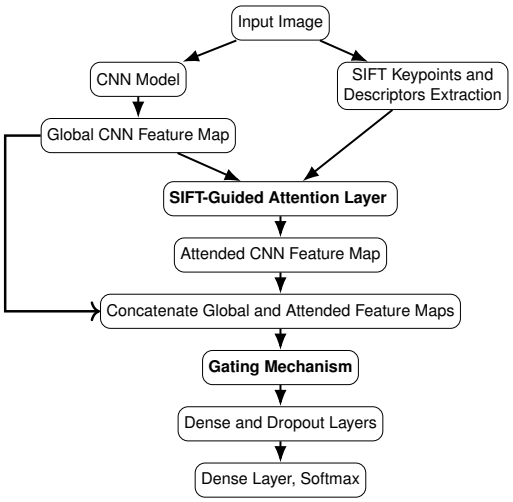


Figure 7: Mid-Level Fusion Approach: Fusion of Local Attended CNN Features and Global CNN Features (LFGF) with Gating Mechanism

with the attended CNN features.

As depicted in Figure 7, the feature map extracted by the RGB branch along with the keypoints and descriptors extracted by the SIFT branch form the input of the attention layer. The attention layer acts as a bridge between the global CNN features and localized SIFT information. It has three key components: *projection layers*, *scaled dot-product attention* (Vaswani et al., 2017), and *weighting and aggregation*.

Projection Layer: The first step involves projecting the CNN features, SIFT descriptors, and keypoints into a common latent space. This is achieved through the use of separate fully connected (dense) layers for each feature type. These projection layers transform the input features into vectors of the same dimensionality, enabling subsequent similarity calculations. Formally, let $\mathbf{F}_{\text{CNN}}$ denote the CNN features, $\mathbf{F}_{\text{SIFT}}$ the SIFT descriptors, and $\mathbf{K}_{\text{SIFT}}$ the SIFT keypoints. The projection layers can be represented as:

$$\mathbf{P}_{\text{CNN}} = W_{\text{CNN}}\mathbf{F}_{\text{CNN}} + b_{\text{CNN}}$$
$$\mathbf{P}_{\text{SIFT}} = W_{\text{SIFT}}\mathbf{F}_{\text{SIFT}} + b_{\text{SIFT}}$$
$$\mathbf{P}_{\text{KP}} = W_{\text{KP}}\mathbf{K}_{\text{SIFT}} + b_{\text{KP}}$$

where W and b are the weights and biases of the respective projection layers, and $\mathbf{P}_{\text{CNN}}, \mathbf{P}_{\text{SIFT}}$, and $\mathbf{P}_{\text{KP}}$ are the projected features.

Scaled Dot-Product Attention (Vaswani et al., 2017): The projected features are next fed into a scaled dot-product attention mechanism. This attention mechanism computes a score that measures the similarity between the SIFT-derived features (descriptors and keypoints) and the CNN features. These

scores determine the importance of each region in the CNN feature map relative to the SIFT keypoints. The similarity scores are computed using a scaled dot-product operation: $Score(i,j) = \frac{\mathbf{P}_{\text{SIFT},i} \cdot \mathbf{P}^T_{CNN,j}}{\sqrt{d}}$, where d is the dimensionality of the projected features, and $\cdot$ denotes the dot product.

The attention weights are then obtained by applying a softmax function to the similarity scores: $\alpha_{ij} = softmax(Score(i,j)) = \frac{\exp(Score(i,j))}{\sum_k \exp(Score(i,k))}$. These attention weights indicate the degree of relevance of each CNN feature region to the SIFT keypoints.

Weighting and Aggregation: Finally, the attention weights are used to modulate the CNN features. The original CNN feature map is element-wise multiplied by the attention weights, effectively highlighting regions deemed important by the SIFT keypoints/descriptors. The refined features, referred to as "*Attended CNN Features*" are computed as follows:

$$\mathbf{F}_{\text{Attended CNN Features}} = \alpha \odot \mathbf{F}_{\text{CNN}}$$

where α represents the attention weights and $\odot$ denotes the element-wise multiplication.
After obtaining the "Attended CNN Features" these are concatenated with the original CNN features to form a comprehensive feature vector. As depicted in Figure 7 our model integrates a *gating mechanism* and *L1 regularization* to reduce redundancy in the concatenated feature map. The gating mechanism selectively combines the original and attended CNN features by learning to scale the importance of each feature through a sigmoid-activated gate, thus enhancing feature discrimination. L1 regularizations are applied to the dense layers projecting the SIFT descriptors and CNN features, as well as the gating layer, to promote sparsity in the learned weights. This encourages the model to utilize a smaller, more informative subset of features, improving generalization and reducing the risk of overfitting.

4 EXPERIMENTAL STUDY

4.1 Experimental Setup

This section evaluates the performance of the proposed fusion strategies on the EuroSAT real-world dataset. The experiments include baseline models, the early fusion model, and the proposed late and mid-level fusion models. Fusion approaches are implemented using both, the shallow CNN described earlier, and a pre-trained, fine-tuned MobileNetV2 model (Qamar and Bawany, 2023). While not the most accurate pre-trained model, MobileNetV2 offers a good trade-off between accuracy and speed. To avoid functional redundancy, we opted for a spatial attention mechanism instead of channel-wise attention in our late fusion approach using MobileNet. This choice allows the network to focus not only on the significance of features across channels (a task already managed by the depthwise separable convolutions) but also on their spatial importance.

Table 1: Accuracy achieved by the studied models

Model	Accuracy
Baseline	
SIFT-NN model	0.619
Shallow CNN	0.845
Fine-tuned MobileNetV2	0.966
Early Fusion	
Shallow CNN	0.887
Fine-tuned MobileNetV2	0.976
Proposed Late Fusion Approach (ADL)	
Shallow CNN	0.911
Fine-tuned MobileNetV2 - SIFT	0.984
Proposed Mid-Level Fusion Approach (LFGF)	
Shallow CNN	0.924
Fine-tuned MobileNetV2	0.985

All models were implemented using Keras and TensorFlow (Abadi et al., 2015). SIFT keypoints and descriptors were extracted using OpenCV (Culjak et al., 2012). To enhance the robustness of our models, we applied common image augmentation techniques, including random flips, random jitters, random rotations, random crop, noise injections for SIFT descriptors, etc.. The EuroSAT images were stratified by land cover class and split into a 70/15/15 training, validation, and test set. Each model was trained for 100 epochs with early stopping and learning rate reduction on plateau strategies.

4.2 Results & Discussion

The results presented in Table 1 confirm the potential benefits of synergizing handcrafted features with learned CNN features for LULC classification. Notably, all the fusion approaches outperform the SIFT-NN and CNN baseline models. The results also show that not all fusion approaches are equally effective. The late fusion approach achieves an improvement of 47.17% over the baseline SIFT-NN and of 7.68% over the baseline CNN, demonstrating the advantage of applying attention mechanisms to dynamically weigh and integrate salient features from both the CNN and SIFT branches before final classification decisions are made. The mid-level fusion approach, which fuses the original CNN-learned feature map with the SIFT-based attended CNN feature map,

Table 2: Accuracy Achieved by Main Existing Approaches

Model	Accuracy
SVM (Thakur and Panse, 2022)	0.509
Random Forest (Thakur and Panse, 2022)	0.567
SIFT-SVM (Helber et al., 2018)	0.701
SIFT-CNN (Ahmed et al., 2024)	0.916
Pretrained VGG19 with TWSVM (Dewangkoro and Arymurthy, 2021)	0.946
SHapley Additive exPlanations (SHAPs) (Temenos et al., 2023)	0.947
Pretrained GoogleNet (Helber et al., 2019)	0.960
Pretrained ResNet50 (Helber et al., 2019)	0.964
ResNet50 pretrained on in-domain datasets (Neumann et al., 2020)	0.992
µ2Net (Gesmundo and Dean, 2022)	0.992
Multi-Task Pretraining with Vision Transformers (ViT) (Wang et al., 2024)	0.992
Our Mid-Level Fusion Approach - Shallow CNN	0.924
Our Mid-Level Fusion Approach - MobileNetV2	0.985

achieves a 49.27% improvement over SIFT-NN and a 9.22% improvement over the baseline CNN. The late and mid-level fusion approaches outperform the more common early fusion approach, which merges information at the initial stages of processing and might discard some feature details before the network can learn their importance.

The improvements achieved by late and mid-level fusion observed with the pre-trained MobileNetV2 are smaller than with the shallow CNN. This is likely because MobileNetV2's pre-trained features already achieve a high baseline performance, leaving less room for enhancement by additional features. However, gains observed across both models demonstrate the generalizability of the proposed fusion approaches.

As shown in Table 2, existing work heavily relies on fine-tuning pre-trained large models such as ResNet50 (Helber et al., 2019; He et al., 2016), Googlenet (*i.e*. InceptionV1,) (Gesmundo and Dean, 2022), and Vision Transformers (ViT) (Wang et al., 2024). These models deliver very high performance, with accuracies ranging from 96.0% to 99.2%. Our proposed mid-level fusion approach with a shallow CNN achieved an accuracy of 92.4%, which surpasses many traditional approaches and is competitive with some pre-trained deep learning approaches. Furthermore, our mid-level fusion with MobileNetV2 reached an accuracy of 98.5%, which is competitive with high-performance models, and only slightly below the top-performing models at 99.2% (Neumann et al., 2020; Wang et al., 2024). Table 2 highlights that, beyond the prevalent fine-tuning of pre-trained large models, alternative approaches such as the integration of handcrafted features deserve exploration.

As mentioned earlier, our mid-level approach fuses two feature maps: the original CNN-learned feature map and a SIFT-based attended CNN feature map. The original feature map captures the broader context, while the attended feature map prioritizes local details. Figure 8 shows the distribution of attention weights in the attended feature map within the mid-level approach. The peaks around zero suggest that the model still relies on the global context provided by the original CNN feature map. The presence of non-zero peaks in attention weights across classes indicates that the mid-level fusion approach effectively utilizes local features captured by SIFT descriptors. This is crucial for enhancing the model's ability to capture fine-grained details that CNNs may not prioritize.

The distribution patterns also reflect the nature of each LULC class, with more complex classes like Industrial showing a broader spread of attention weights. This indicates a more nuanced use of local features. Homogeneous classes like Forest and SeaLake show a narrow distribution, suggesting a consistent pattern of local feature importance, aligning with their more uniform textures.

The analysis of attention weight distributions highlights the potential of the mid-level fusion approach for integrating local details captured by SIFT descriptors with the global context learned by CNNs. This approach demonstrably enhances the model's ability to capture fine-grained information crucial for accurate LULC classification. However, further investigation is necessary to determine the generalizability of these findings across various datasets and LULC tasks. Additionally, exploring alternative attention mechanisms or feature extraction techniques might be beneficial for capturing even more nuanced local features or handling situations where SIFT descriptors might not be optimal.

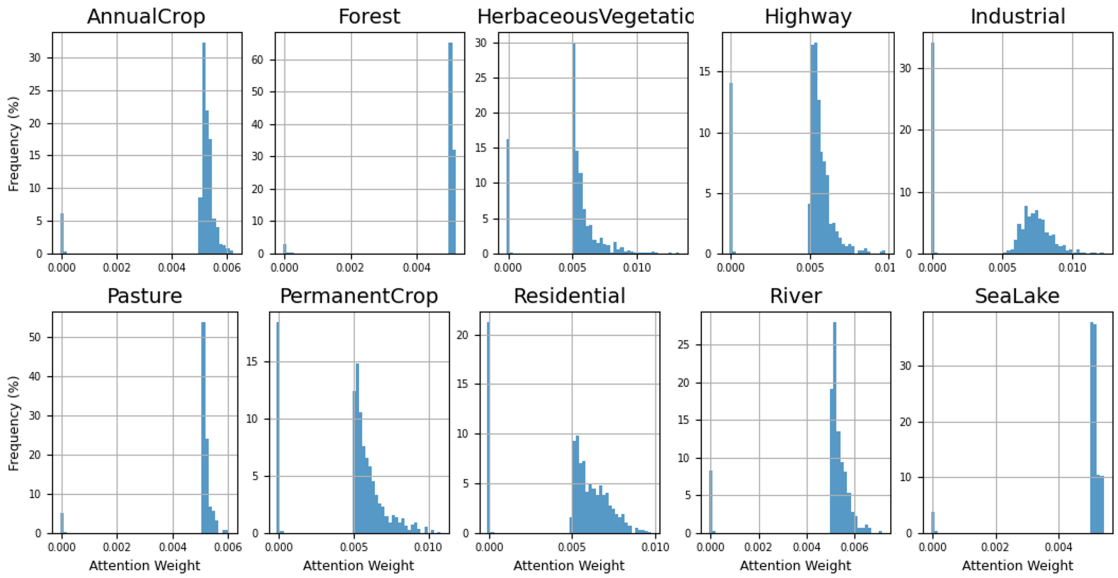


Figure 8: Distribution of Attention Weights (LFGF-CNN)

5 CONCLUSION AND FUTURE WORK

This paper investigated the synergistic integration of handcrafted SIFT descriptors with CNN-learned features for improved LULC classification accuracy. We compared three fusion strategies: early, late, and mid-level fusion. Late fusion dynamically weighs salient features from both modalities before classification. Mid-level fusion further refines this by using a custom SIFT-guided attention mechanism, selectively amplifying detailed features while preserving the rich CNN features. Experiments on real-world data showed that late and mid-level fusion outperform the conventional early fusion approach, demonstrating their efficacy in capturing both fine-grained local details and broader scene context.

The encouraging results of fusion approaches pave the way for several research directions. Moving forward, we plan to delve deeper into the realm of attention-based methods and dynamic fusion approaches. We also envision exploring the application of the proposed method to land-use change detection by analyzing time series data. Another interesting direction involves investigating the generalizability and adaptability of our proposed fusion approaches by applying them to various data-intensive tasks beyond LULC classification.

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citron, C., Corrado, G., Davis, A., Dean, J., Devin, M., and et. al. (2015). Tensorflow: Large-scale machine learning on heterogeneous systems.

Ahmed, V. A., Jouini, K., Tuama, A., and Korbaa, O. (2024). A fusion approach for enhanced remote sensing image classification. In Radeva, P., Furnari, A., Bouatouch, K., and de Sousa, A. A., editors, *Proceedings of the 19th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, VISIGRAPP 2024, Volume 2: VISAPP, Rome, Italy, February 27-29, 2024*, pages 554–561. SCITEPRESS.

Chen, S. and Tian, Y. (2015). Pyramid of spatial relatons for scene-level land use classification. *IEEE Trans. Geosci. Remote Sens.*, 53(4):1947–1957.

Cheng, G., Xie, X., Han, J., Guo, L., and Xia, G. S. (2019). Remote sensing image scene classification meets deep learning: Challenges, methods, benchmarks, and opportunities. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, 13(X):3735–3756.

Culjak, I., Abram, D., Pribanic, T., Dzapo, H., and Cifrek, M. (2012). A brief introduction to opencv. In *35th International Convention MIPRO*, page 1725–1730.

Dewangkoro, H. I. and Arymurthy, A. M. (2021). Land use and land cover classification using cnn, svm, and channel squeeze & spatial excitation block. *IOP Conf. Ser. Earth Environ. Sci.*, 704(1).

Gesmundo, A. and Dean, J. (2022). An evolutionary approach to dynamic introduction of tasks in large-scale multitask learning systems.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of*

the IEEE conference on computer vision and pattern recognition, pages 770–778.

Helber, P., Bischke, B., Dengel, A., and Borth, D. (2018). Introducing eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. In *IGARSS*, page 204–207.

Helber, P., Bischke, B., Dengel, A., and Borth, D. (2019). Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, 12(7):2217–2226.

Hu, J., Shen, L., and Sun, G. (2018). Squeeze-and-excitation networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7132–7141.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.

Neumann, M., Pinto, A. S., Zhai, X., and Houlsby, N. (2020). In-domain representation learning for remote sensing. *CoRR*, abs/1911.06721.

Qamar, T. and Bawany, N. Z. (2023). Understanding the black-box: towards interpretable and reliable deep learning models. *PeerJ Comput. Sci.*, 9.

Steiner, A., Kolesnikov, A., Zhai, X., Wightman, R., Uszkoreit, J., and Beyer, L. (2021). How to train your vit? data, augmentation, and regularization in vision transformers. *CoRR*, abs/2106.10270.

Temenos, A., Temenos, N., Kaselimi, M., Doulamis, A., and Doulamis, N. (2023). Interpretable deep learning framework for land use and land cover classification in remote sensing using shap. *IEEE Geosci. Remote Sens. Lett.*, 20:1–5.

Thakur, R. and Panse, P. (2022). Classification performance of land use from multispectral remote sensing images using decision tree, k-nearest neighbor, random forest and support vector machine using eurosat da. *Orig. Res. Pap. Int. J. Intell. Syst. Appl. Eng. IJISAE*, 2022(1s):67–77. [Online]. Available: https://github.com/phelber/EuroSAT.

Tianyu, Z., Zhenjiang, M., and Jianhu, Z. (2018). Combining cnn with hand-crafted features for image classification. In *2018 14th IEEE International Conference on Signal Processing (ICSP)*, pages 554–557.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Wang, D., Zhang, J., Xu, M., Liu, L., Wang, D., Gao, E., Han, C., Guo, H., Du, B., Tao, D., and Zhang, L. (2024). Mtp: Advancing remote sensing foundation model via multi-task pretraining.

Xia, H. and Liu, C. (2019). Remote sensing image deblurring algorithm based on wgan. In Liu, X., Mrissa, M., Zhang, L., Benslimane, D., Ghose, A., Wang, Z., Bucchiarone, A., Zhang, W., Zou, Y., and Yu, Q., editors, *Service-Oriented Computing – ICSOC 2018 Workshops*, pages 113–125, Cham. Springer International Publishing.

.

B.2 Augmentation-Based Ensemble Learning for Stance and Fake News Detection

Ilhem Salah, Khaled Jouini & Ouajdi Korbaa

14^{th} International Conference on Computational Collective Intelligence (ICCCI). 2022.

Lien : https://link.springer.com/chapter/10.1007/978-3-031-16210-7_3

CORE Rank B

ICORE Conference Portal

Back to search

International Conference on Computational Collective Intelligence

Acronym: ICCCI

DBLP Source: https://dblp.uni-trier.de/db/conf/iccci

Source: CORE2023

Rank: B

Field Of Research: 4602 - Artificial intelligence (h-index) (citation)

Field Of Research: 4606 - Distributed computing and systems software (h-index) (citation)

Source: CORE2021

Rank: B

Field Of Research: 4602 - Artificial intelligence

Field Of Research: 4606 - Distributed computing and systems software

Request for rank B accepted (Data 1) (Decision)

Source: CORE2020

Rank: C

Field Of Research: 4602 - Artificial intelligence

Field Of Research: 4606 - Distributed computing and systems software

Source: CORE2018

Rank: C

Field Of Research: 0801 - Artificial Intelligence and Image Processing†

Source: CORE2017

Rank: C

Field Of Research: 0801 - Artificial Intelligence and Image Processing†

Augmentation-Based Ensemble Learning For Stance and Fake News Detection

Ilhem Salah[0000−0002−3375−3637], Khaled Jouini[0000−0001−5049−4238], and Ouajdi Korbaa[0000−0003−4462−1805]

MARS Research Lab LR17ES05, ISITCom, University of Sousse, H. Sousse, 4011, Tunisia
ilhemsalah53@gmail.com, khaled.jouini@isitc.u-sousse.tn, ouajdi.korbaa@centraliens-lille.org

Abstract. Data augmentation is an unsupervised technique used to generate additional training data by slightly modifying already existing data. Besides preventing data scarcity, one of the main interest of data augmentation is that it increases training data diversity, and hence improves models' ability to generalize to unseen data. In this work we investigate the use of text data augmentation for the task of stance and fake news detection.
In the first part of our work, we explore the effect of various text augmentation techniques on the performance of common classification algorithms. Besides identifying the best performing (classification algorithm, augmentation technique) pairs, our study reveals that the motto "*the more, the better*" is the wrong approach regarding text augmentation and that there is no *one-size-fits-all* text augmentation technique.
The second part of our work leverages the results of our study to propose a novel augmentation-based, ensemble learning approach that can be seen as a mixture between stacking and bagging. The proposed approach leverages text augmentation to enhance base learners' diversity and accuracy, ergo the predictive performance of the ensemble. Experiments conducted on two real-world datasets show that our ensemble learning approach achieves very promising predictive performances.

Keywords: Stance and Fake News Detection · Text Augmentation · Ensemble Learning · Fake News Challenge.

1 Introduction

In the era of the Internet and social media, where a myriad of information of various types is instantly available and where any point of view can find an audience, access to information is no longer an issue, and the key challenges are veracity, credibility, and authenticity. The reason for this is that any user can readily gather, consume, and break news, without verification, fact-checking, or third-party filtering. As revealed by several recent studies, fake news and misinformation are prone to spread substantially faster, wider, and deeper than genuine news and real information [21, 8].

By directly influencing public opinions, major political events, and societal debates, fake news has become the scourge of the digital era, and combating it has become a dire need. The identification of fake news is however very challenging, not only from a machine learning and Natural Language Processing (NLP) perspective, but also sometimes for the most experienced journalists [18]. That is why the scientific community approaches the task from a variety of angles and often breaks down the process into independent sub-tasks. A first practical step towards automatic fact-checking and fake news detection is to estimate the opinion or the point of view (*i.e. stance*) of different news sources regarding the same topic or claim [18]. This (sub-) task, addressed in recent research as *stance detection*, was popularized by the Fake News Challenge - Stage 1 (or FNC-1) [18], which compares article bodies to article headlines and determines if a body agrees, disagrees, discusses or is unrelated to the claim of a headline.

In this paper we propose a novel *Augmentation-based Ensemble learning* approach for stance and fake news detection. Data augmentation refers to techniques used to create new training data by slightly modifying available labelled data. Besides preventing data scarcity, one of the main interest of data augmentation is that it increases training data diversity, and hence helps to improve models' ability to generalize to unseen data [11]. Data augmentation is extensively used in Computer Vision (CV) where it is considered as one of the anchors of good predictive performance. Despite promising advances, data augmentation remains however less explored in NLP where it is still considered as the "cherry on the cake" which provides a steady but limited performance boost [23].

Ensemble learning combines the knowledge acquired by base learners to make a consensus decision which is supposed to be superior to the one attained by each base learner alone [27]. Research on ensemble learning proves that the greater are the skills and the diversity of base learners, the better are the accuracy and the generalization ability of the ensemble [27]. In this work we leverage text data augmentation to enhance both, the diversity and the skills of base learners, ergo the accuracy of the ensemble.

The main contributions of our work are therefore: (*i*) an extensive experimental study on the effect of different text data augmentation techniques on the performance of common classification algorithms in the context of stance and fake news detection. Our study provides insights for practitioners and researchers on text data augmentation and the best performing (data augmentation technique, classification algorithm) pairs; and (*ii*) a novel augmentation-based ensemble learning approach, which is a mixture of stacking and bagging.

The remainder of this paper is organized as follows. Section 2 outlines the main steps we followed to vectorize text and reduce dimensionality. Section 3 exposes the key motifs of data augmentation and the text augmentation techniques adopted in our work. Section 4 details the architecture of our novel augmentation-based ensemble learning. Section 5 briefly reviews existing work on stance and fake news detection. Section 6 presents an experimental study on two real-world fake news datasets and discusses the main results and findings. Finally, section 7 concludes the paper.

2 Text as Vectors

2.1 Pre-Processing and Feature Extraction

Machine Learning (ML) algorithms operate on numerical features, expecting input in the form of a matrix where rows represent instances and columns features. Raw news texts have therefore to be transformed into feature vectors before feeding into ML algorithms [9]. In our work, we first eliminated stop words and reduced words to their roots (*i.e.* base words) by stemming them using Snowball Stemmer from the NLTK library [16]. We next vectorized the corpus with a TF-IDF (*Term Frequency – Inverse Document Frequency*) weighting scheme and generated a term-document matrix.

TF–IDF is computed on a per-term basis, such that the relevance of a term to a text is measured by the scaled frequency of the appearance of the term in the text, normalized by the inverse of the scaled frequency of the term in the entire corpus. Despite its simplicity and its wide-spread use, the TF-IDF scheme has two severe limitations: (*i*) TF-IDF does not capture the co-occurrence of terms in the corpus and makes no use of semantic similarities between words. Accordingly, TF-IDF fails to capture some basic linguistic notions such as synonymy and homonymy; and (*ii*) The term-document matrix is high dimensional and is often noisy, redundant, and excessively sparse. The matrix is thus subject to the curse of dimensionality: as the number of features is large, poor generalization is to be expected.

2.2 Dimensionality Reduction

Latent Semantic Analysis (LSA) [3] is an unsupervised statistical topic modeling technique, overcoming some of the limitations of TF-IDF. As other topic modeling techniques, such as LDA (Latent Dirichlet Allocation [2]), LSA is based on the assumptions that: (*i*) each text consists of a mixture of topics; and (*ii*) each topic consists of a set of (weighted) terms that regularly co-occur together. Put differently, the basic assumption behind LSA is that words that are close in meaning, appear in similar contexts and form a "hidden topic". The basic intuition behind LSA is to represent words that form a topic not as separate dimensions, but by a single dimension. LSA represents thus texts by "semantic" or "topic" vectors, based on the words that these texts contain and the set of weighted words that form each of the topics.

To uncover the latent topics that shapes the meaning of texts, LSA performs a Singular Value Decomposition (SVD) on the document-term matrix (*i.e.* decomposes it into a separate text-topic matrix and a topic-term matrix). Formally, SVD decomposes the term-document matrix $A_{t\times n}$, with t the number terms and d the number of documents, into the product of three different matrices: orthogonal column matrix, orthogonal row matrix and one singular matrix.

$$A_{t\times n} = U_{t\times n} S_{n\times n} D^T_{n\times d} \tag{1}$$

where $n = min(t, d)$ is the rank of A. By restricting the matrices T, S and D to their first $k < n$ rows, we obtain the matrices $T_{t\times k}$, $S_{k\times k}$ and $D_{d\times k}$, and hence obtain k-dimensional text vectors. From a practical perspective the key ask is to determine k, which would be reasonable for the problem (*i.e.* without major loss). In our work we used the transformer TruncatedSVD from sklearn [17]. As in [12] we set the value of k to 100D. The experimental study conducted in [12] showed that using LSA (with k set to 100D) instead of TF-IDF allows a substantial performance improvement for the tasks of stance and fake news detection.

3 Text Data Augmentation

Data augmentation aims at synthesizing new training instances that have the same ground-truth labels as the instances that they originate from [30]. Data augmentation has several well-known benefits: (*i*) preventing overfitting by improving the diversity of training data; (*ii*) preventing data scarcity by providing a relatively easy and inexpensive way to collect and label data; (*iii*) helping resolve class imbalance issues; and (*iv*) increasing the generalization ability of the obtained model.

The success of data augmentation in Computer Vision has been fueled by the ease of designing semantically invariant transformations (*i.e.* label-preserving transformations), such as rotation, flipping, etc... While recent years witnessed significant advancements in the design of transformation techniques, text augmentation remains less explored and adopted in NLP than in CV. This is mainly due to the intrinsic properties of textual data (*e.g.* polysemy), which make defining label-preserving transformations much harder [23]. In the sequel we mainly focus on off-the-shelf text augmentation techniques and less on techniques that are still in the research phase, waiting for large-scale testing and adoption. For a more exhaustive survey on text augmentation techniques, we refer the reader to [11, 1, 28].

3.1 Masked Language Models

The main idea behind Masked Language Models (MLMs), such as BERT [4], is to mask words in sentences and let the model predict the masked words. BERT, which is a pretrained multi-layer bidirectional transformer encoder, has the ability to predict masked words based on the bidirectional context (*i.e.* based on its left and right surrounding words). In contrast with other context-free models such as GLOVE and Word2Vec, BERT alleviates the problem of ambiguity since it considers the whole context of a word.

BERT is considered as a breakthrough in the use of ML for NLP and is widely used in a variety of tasks such as classification, Question/Answering, and Named Entity Recognition [22]. Inspired by the recent work of [22, 11], we use BERT as an augmentation technique. The idea is to generate new sentences by randomly masking words and replacing them by those predicted by BERT.

3.2 Back-translation (*a.k.a.* Round-trip translation)

Back-Translation is the process of translating a text into another language, then translating the new text back into the original language. Back-translation is one of the most popular means of paraphrasing and text data augmentation [15]. Google Cloud Translation API, used in our work to translate sentences to French and back, is considered as the most common tool for back-translation [11].

3.3 Synonym (*a.k.a.* Thesaurus-based augmentation)

The synonym technique, also called lexical substitution with dictionary, was until recently the most widely (and for a long time the only) augmentation technique used for textual data classification. As suggested by its name, the Synonym technique replaces randomly selected words with their respective synonyms. The types of words that are candidates for lexical substitution are: adverbs, adjectives, nouns and verbs.

The synonyms are typically taken from a lexical database (*i.e.* dictionary of synonyms). WordNet [6], used in our work for synonym replacement, is considered as the most popular open-source lexical database for the English language [11].

3.4 TF-IDF based Insertion and substitution

The intuition behind these two noising-based techniques is that uninformative words (*i.e.* having low TF-IDF scores) should have no or little impact on classification. Therefore, the insertion of words having low TF-IDF scores (at random positions) should preserve the label associated with a text, even if the semantics are not preserved. An alternate strategy is to replace randomly selected words with words having the same low TF-IDF scores (TF-IDF based substitution).

Section 6 presents an extensive study on the effect of the aforementioned augmentation techniques on the preredictive performance of ten common classification algorithms, namely, Decision Tree (DT), Support Vector Machine (SVM), Adaptive Boosting (AdaBoost), Random Forest (RF), eXtreme Gradient Boosting (XGBoost), Bagged Random Forests (Bagged RF), Light Gradient Boosting Machine (LightGBM), Gradient Boosting (GradBoost), Logistic Regression (LR), and Naive Bayes (NB). Moreover, in contrast with existing work, where text augmentation is considered as an auxiliary technique, our novel augmentation-based ensemble approach presented in next section, goes further and let augmentation shape the entire learning process.

4 Augmentation-Based Ensemble Learning

4.1 Diversity and Skillfulness in Ensemble Learning

Ensemble Learning finds its origins in the "Wisdom of Crowds" theory [26]. The "Wisdom of Crowds" theory states that the collective opinion of a group

of individuals can be better than the opinion of a single expert, provided that the aggregated opinions are diverse (*i.e.* diversity of opinion) and that each individual in the group has a minimum level of competence (*e.g.* better than a random guess). Similarly, Ensemble Learning combines the knowledge acquired by a group of base learners to make a consensus decision which is supposed to be superior to the one reached by each of them separately [27]. Research on Ensemble Learning proves that the greater are the skills and the diversity of base models, the better is the generalization ability of the ensemble model [27]. Alternatively stated, to generate a good ensemble model, it is necessary to build base models that are, not only skillful, but also skillful in a different way from one another.

Bagging and stacking are among the main classes of parallel ensemble techniques. Bagging (*i.e.* Bootstrap aggregating) involves training multiple instances of the same classification algorithm, then combining the predictions of the obtained models through hard or soft voting. To promote diversity, base learners are trained on different subsets of the original training set. Each subset is typically obtained by drawing random samples with replacement from the original training set (*i.e.* bootstrap samples).

Stacking (*a.k.a.* stacked generalization) involves training a learning algorithm (*i.e.* meta-classifier) to combine the predictions of several heterogeneous learning algorithms, trained on the same training data. The most common approach to train the meta-model is via k-fold cross-validation. With the k-fold cross-validation, the whole training dataset is randomly split (without replacement) into independent equal-sized k-folds. $k-1$ folds are then used to train each of the base models and the k^{th} fold (holdout fold) is used to collect the predictions of base models on unseen data. The predictions made by base models on the holdout fold, along with the expected class labels, provide the input and the output pairs used to train the meta-model. This procedure is repeated k times. Each time a different fold acts as the holdout fold while the remaining folds are combined and used for training the base models.

4.2 Novel Augmentation Based Approach

As mentioned earlier, in conventional stacking base learners are trained on the same dataset and diversity is achieved by using heterogeneous classification algorithms. As depicted in figure 1, the classical approach for combining augmentation and stacking, is to: (*i*) apply one or several augmentation techniques to the original dataset, (*ii*) fuse the original dataset with data obtained through augmentation; and (*iii*) train base learners on the fused dataset.

In our work we adopt a different approach and train heterogeneous algorithms on different data to further promote diversity. More specifically, through an extensive experimental study (Section 6), we first identify the most accurate (augmentation technique, classification algorithm) pairs. Our meta-model is then trained on the predictions made by the most accurate pairs, using a stratified k-fold cross-validation. Figure 2 depicts the overall architecture of the proposed augmentation-based ensemble learning.

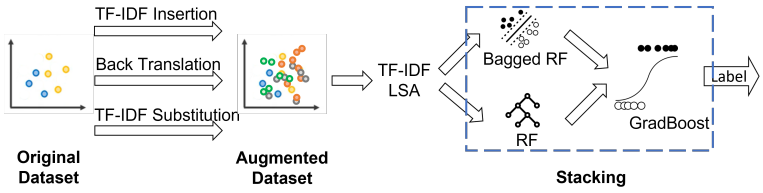


Fig. 1. Conventional approach for combining augmentation and stacking

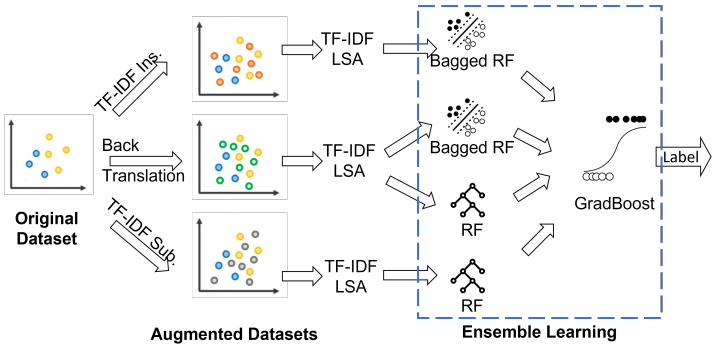


Fig. 2. Novel Augmentation-Based Ensemble Learning Approach

Our augmentation-based ensemble learning approach, can be seen as a mixture between stacking and bagging. In contrast with Bagging and like Stacking, we use an ensemble of heterogeneous learning algorithms. In contrast with stacking and like Bagging, base learners are trained on different datasets, to further promote diversity. However, unlike Bagging the considered datasets are not obtained through bootstrap sampling. Instead, they are obtained by combining the original training data with the data obtained by applying one of the text augmentation techniques. Finally, like in conventional Stacking, the meta-model is trained using a stratified K-fold cross-validation.

5 Related Work

Salient stance and fake news detection approaches adopt a wide range of different features (*e.g.*, context-based, content-based), classifiers, and learning tactics (*e.g.* stacking, bagging, etc.) [5]. Due to the lack of space, we mainly focus hereafter on ensemble approaches and on approaches that rely on content-based features. We suggest readers to refer to surveys and retrospectives on recent challenges [7, 10] for a more comprehensive overview of the current state of research.

The authors of the fake news challenge (FNC-1) [25], released a simple baseline model for the stance detection task. The proposed model achieves an F1-score of 79.53% and uses a gradient boosting (GradBoost) classifier on global co-occurrence, polarity and refutation features. The three best performing systems in the FNC-1 competition were "SOLAT in the SWEN" [20], "Team Athene"

[7] and "UCL Machine Reading" (UCLMR) [19]. "SOLAT in the SWEN" won the competition using an ensemble approach based on a 50/50 weighted average between gradient-boosted decision trees and a Convolutional Neural Network (CNN). The proposed system is based on several features: word2vec pretrained embeddings, TF-IDF, Single Value Decomposition and Word Count. The convolutional network uses pre-trained Word2Vec embeddings passed through several convolutional layers followed by three fully-connected layers and a final softmax layer for classification.

[7], the second place winner, used an ensemble composed of 5 Multi-Layer Perceptrons (MLPs), where labels are predicted through hard voting. The system of UCLMR [19], placed third, used an MLP classifier with one hidden layer of 100 units and a softmax layer for classification. In the same vein as [7], [14] uses a hard voting classifier. The ensemble is composed of three base learners, namely, MLP, Logistic Regression (LR) and X-Gradient Boosting (XGBoost). [14] experimented their approach on the dataset LIAR proposed by [29].

Recently, other published work used FNC-1 in their experiments. [5] constructed a stance detection language model by performing transfer learning on a RoBERTa deep bidirectional transformer language model. [5] leverages bidirectional cross-attention between claim-article pairs via pair encoding with self-attention. The work of [12], which is the closest to the spirit of our work, uses LSA for dimensionality reduction and a stacking-based ensemble having five base learners: GradBoost, Random Forest (RF), XGBoost, Bagging and Light Gradient Boosting Machine (Lightgbm). Besides, [12] compared LDA and LSA and found that LSA yields better accuracy. The authors in [12] experimented their approach on FNC-1 and FNN datasets.

It is worth noticing that in all the aforementioned studies, ensemble approaches yielded better results than those attained by their contributing base learners. On the other hand, despite the substantial potential improvement that text augmentation can carry out, to the best of our knowledge there exists no previous work on stance and fake news detection that compares text augmentation techniques and uses text augmentation in conjunction with ensemble learning.

6 Experimental Study

6.1 Tools & Datasets

Our system was implemented using NLTK [16] for text preprocessing, nlpaug[13] for text augmentation, SciKit-Learn (version 0.24.2) [17] for classification and Beautiful Soup for web scraping. A stratified 10-fold cross-validation was used for model fusion. The Li & al. approach was implemented as described in [12]. The experimental study was conducted without any special tuning. A large number of experiments have been performed to show the accuracy and the effectiveness of our augmentation-based ensemble learning. Due to the lack of space, only few results are presented herein.

As there are no agreed-upon benchmark datasets for stance and fake news detection [12], we used two publicly available and complementary datasets: FNC-

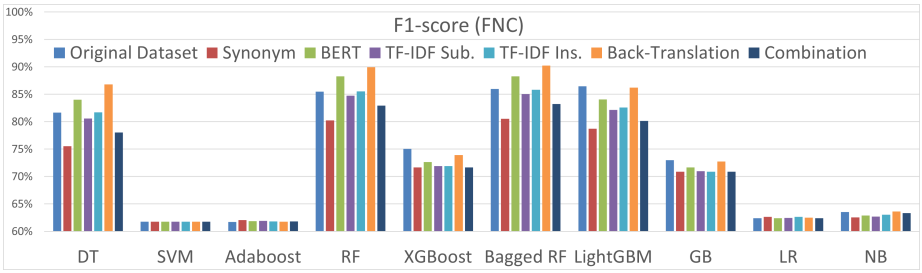


Fig. 3. F1-scores on FNC with and without text augmentation

1 [18] and FNN (*i.e.* FakeNewsNet) [24]. FNC was released to explore the task of stance detection in the context of fake news detection. Stance detection is a multinomial classification problem, where the relative stance of each headline-article pair has to be classified as either: *Agree* if the article agrees with the headline claim, *Disagree* if the article disagrees with the claim, *Discuss* if the article is related to the claim, but takes no position on the subject, and *Unrelated* if the content of the article is unrelated to the claim. The FNC-1 dataset consists of approximately 50k headline-article pairs in the training set and 25k pairs in the test set. FNN data was collected from two fact-checking websites (*i.e.* GossipCop and PolitiFact) containing news contents, along with context information. In comparison with FNN, FNC-1 provides fewer data features (4 vs. 13 features), but more data ($\approx$ 75k vs. $\approx$ 997).

6.2 Results and discussion

We ran our experiments with three objectives in mind: (*i*) identify the best performing (*Augmentation technique*, *Classifier*) pairs; (*ii*) quantify the actual performance improvement allowed by each text augmentation technique; and (*iii*) evaluate the effectiveness of our augmentation-based ensemble approach.

Best performing pairs Figure 3 (resp. 4), reports the F1-scores obtained on FNC (resp. FNN). The results presented in these charts allow to draw important conclusions regarding text augmentation:

1. *Text augmentation does not always improve predictive performance.* This can be especially observed for SVM, LightGBM, GradBoost (figure 3) and AdaBoost (figure 4), where the F1-scores on the original dataset are higher than to those obtained on the augmented datasets;
2. *There is no one-size-fits-all augmentation technique that performs well in all situations.* As depicted in figures 3 and 4, an augmentation technique may perform well when combined with a classification algorithm and poorly when combined with another. This is the case for example for the "Synonym" technique which yields the highest F1-score when combined with Adaboost and the lowest score when used with Naive Bayes (Figure 3).

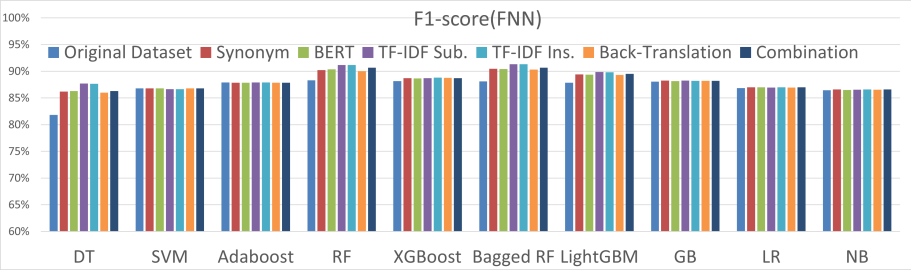


Fig. 4. F1-scores on FNN with and without text augmentation

It is worth noting that even if BERT doesn't achieve the highest F1-scores, it provides a steady performance improvement for almost all classifiers;

3. *The motto "the more, the better" is the wrong approach regarding text augmentation and targeted approaches allow often better results.* This can be observed in figures 3 and 4, where in almost all cases, combining all augmentation techniques does not yield the best F1-scores.

As shown in figure 3, the pairs (Back-translation, Bagged RF) and (Back-translation, RF) yield the highest F1-scores on FNC and increase substantially the predictive performances ($\approx$ + 4.16% in comparison with the highest F1-Score that can be achieved without text augmentation). Similarly, as shown in figure 4, the pairs (Substitution TF-IDF, RF) and (Insertion TF-IDF, Bagged RF) yield the best F1-scores on the dataset FNN ($\approx$ + 5.87%).

Augmentation-Based Ensemble Learning As previously stated, base learners' diversity and competency are the two key success factors of any ensemble learning approach. Our ensemble approach leverages text augmentation to enhance both. Figure 2 depicts our classification model which is a mixture of stacking and bagging. In our model, we use Bagged RF and Random Forest (RF) as base classifiers and GradBoost as meta-classifier. As depicted in figure 2, each of the base classifiers is trained on a dataset composed of the original dataset and the data obtained by applying one of the augmentation techniques. The choice of the (classifier, augmentation technique) pairs was driven by the experimental study conducted in subsection 6.2. We compare our model to a more classical stacking approach, where all base classifiers are trained on the same dataset, consisting of the original dataset and the data obtained by applying one of the augmentation techniques (figure 1). We also compare our model to the approach of [12], which is one of the state-of-the-art approaches that uses LSA, stacking-based ensemble learning and K-fold cross-validation. Table 1 synthesizes the predictive performances achieved by each approach.

As reported in Table 1, the use of text augmentation allows better performances than those achieved by [12] in almost all situations. On the other hand, except for the Synonym technique over the FNC dataset, our model outperforms the classical approach in all situations. Overall, our stacking approach achieves

Table 1. F1-scores achieved by conventional stacking, [12] and the proposed approach

Model	FNC	FNN
(Insertion TF-IDF, Stacking)	85,58%	**90,92%**
(Substitution TF-IDF, Stacking)	84,57%	90,43%
(Back-Translation, Stacking)	**90,31%**	89,80%
(BERT, Stacking)	87,93%	90,26%
(Synonym, Stacking)	80,71%	90,28%
(Combination, Stacking)	83,11%	90,73%
Li & al. [12]	83,72%	88,45%
Proposed approach	**90,15%**	**91,07%**

an increase in F1-score of 7,72% (resp. 7,54%) over FNC (resp. FNN) when compared to [12].

7 Conclusion

Combating fake news on social media is a pressing need and a daunting task. Most of the existing approaches on fake news detection, focus on using various features to identify those allowing the best predictive performance. Such approaches tend to undermine the generalization ability of the obtained models.

In this work, we investigated the use of text augmentation in the context of stance and fake news detection. In the first part of our work, we studied the effect of text augmentation on the performance of various classification algorithms. Our experimental study quantified the actual contribution of data augmentation and identified the best performing (classifier, augmentation technique) pairs. Besides, our study revealed that the motto "the more, the better" is the wrong approach regarding text augmentation and that there is no one-size-fits-all augmentation technique. In the second part of our work, we proposed a novel augmentation-based ensemble learning approach. The proposed approach is a mixture of bagging and stacking and leverages text augmentation to enhance the diversity and the performance of base classifiers. We evaluated our approach using two real-world datasets. Experimental results show that it is more accurate than state-of-art methods.

As a part of our future work, we intend to explore the use of a multimodal data augmentation that involves linguistic and extra linguistic features. We also intend to explore the detection of fake news from streams under concept drifts.

References

1. Andrea Stevens Karnyoto, Chengjie Sun, B.L., Wang, X.: Augmentation and heterogeneous graph neural network for aaai2021-covid-19 fake news detection. International journal of machine learning and cybernetics p. 13 (2022). https://doi.org/10.1007/s13042-021-01503-5
2. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of Machine Learning Research **3**, 993–1022 (Mar 2003)

3. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. Journal of the American Society for Information Science **41**(6), 391–407 (1990)
4. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. CoRR **abs/1810.04805** (2018)
5. Dulhanty, C., Deglint, J.L., Daya, I.B., Wong, A.: Taking a stance on fake news: Towards automatic disinformation assessment via deep bidirectional transformer language models for stance detection. CoRR **abs/1911.11951** (2019)
6. Fellbaum, C.: Wordnet and wordnets. In: Barber, A. (ed.) Encyclopedia of Language and Linguistics, pp. 2–665. Elsevier (2005)
7. Hanselowski, A., P.V.S., A., Schiller, B., Caspelherr, F., Chaudhuri, D., Meyer, C.M., Gurevych, I.: A retrospective analysis of the fake news challenge stance-detection task (2018)
8. Hsu, C.C., Ajorlou, A., Jadbabaie, Ali, P.: News sharing, and cascades on social networks. https://ssrn.com/abstract=3934010 or http://dx.doi.org/10.2139/ssrn.3934010 (December 2021), [Accessed: 2022-01-05]
9. Jouini, K., Maaloul, M.H., Korbaa, O.: Real-time, cnn-based assistive device for visually impaired people. In: 2021 14th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI). pp. 1–6 (2021)
10. Khan, J.Y., Khondaker, M.T.I., Afroz, S., Uddin, G., Iqbal, A.: A benchmark study of machine learning models for online fake news detection. Machine Learning with Applications **4**, 100032 (2021). https://doi.org/https://doi.org/10.1016/j.mlwa.2021.100032, https://www.sciencedirect.com/science/article/pii/S266682702100013X
11. Li, B., Hou, Y., Che, W.: Data augmentation approaches in natural language processing: A survey. CoRR **abs/2110.01852** (2021), https://arxiv.org/abs/2110.01852
12. Li, S., Ma, K., Niu, X., Wang, Y., Ji, K., Yu, Z., Chen, Z.: Stacking-based ensemble learning on low dimensional features for fake news detection. In: 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS) (2019). https://doi.org/10.1109/HPCC/SmartCity/DSS.2019.00383
13. Ma, E.: NLP Augmentation. https://github.com/makcedward/nlpaug (2019), [Accessed: 2021-05-15]
14. Mahabub, A.: A robust technique of fake news detection using ensemble voting classifier and comparison with other classifiers. SN Applied Sciences **2** (04 2020). https://doi.org/10.1007/s42452-020-2326-y
15. Marivate, V., Sefara, T.: Improving short text classification through global augmentation methods. CoRR **abs/1907.03752** (2019), http://arxiv.org/abs/1907.03752
16. NLTK.org: Natural Language Toolkit. https://github.com/nltk/nltk, [Accessed: 2021-05-15]
17. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
18. Pomerleau, D., Rao, D.: The fake news challenge: Exploring how artificial intelligence technologies could be leveraged to combat fake news. http://www.fakenewschallenge.org/ (2017), [Accessed: 2021-12-15]

19. Riedel, B., Augenstein, I., Spithourakis, G.P., Riedel, S.: A simple but tough-to-beat baseline for the Fake News Challenge stance detection task. CoRR **abs/1707.03264** (2017), http://arxiv.org/abs/1707.03264
20. Sepúlveda Torres, R., Vicente, M., Saquete, E., Lloret, E., Sanz, M.: Headlinestancechecker: Exploiting summarization to detect headline disinformation. Journal of Web Semantics **71**, 100660 (09 2021). https://doi.org/10.1016/j.websem.2021.100660
21. Serrano, E., Iglesias, C.A., Garijo, M.: A survey of twitter rumor spreading simulations. In: Int. Conf. Computational Collective Intelligence (ICCCI'15). pp. 113–122. Springer International Publishing, Cham (2015)
22. Shi, L., Liu, D., Liu, G., Meng, K.: Aug-bert: An efficient data augmentation algorithm for text classification. In: Liang, Q., Wang, W., Liu, X., Na, Z., Jia, M., Zhang, B. (eds.) Communications, Signal Processing, and Systems. pp. 2191–2198. Springer Singapore, Singapore (2020)
23. Shorten, C., Khoshgoftaar, T., Furht, B.: Text data augmentation for deep learning. Journal of Big Data **8** (07 2021). https://doi.org/10.1186/s40537-021-00492-0
24. Shu, K.: FakeNewsNet (2019). https://doi.org/10.7910/DVN/UEMMHS, [Accessed: 2021-12-15]
25. Slovikovskaya, V.: Transfer learning from transformers to fake news challenge stance detection (fnc-1) task. In: Proceedings of the 12th Language Resources and Evaluation Conference. pp. 1211–1218. European Language Resources Association (2019), https://www.aclweb.org/anthology/2020.lrec-1.152
26. Surowiecki, J.: The Wisdom of Crowds. Anchor Books, 1st edn. (2005)
27. Suting, Y., Ning, Z.: Construction of structural diversity of ensemble learning based on classification coding. In: 2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC). vol. 9, pp. 1205–1208 (2020). https://doi.org/10.1109/ITAIC49862.2020.9338807
28. Tesfagergish Senait Gebremichael, Robertas Damaševičius, J.K.D.: Deep fake recognition in tweets using text augmentation, word embeddings and deep learning. In: Computational Science and Its Applications – ICCSA 2021: 21st International Conference. pp. 113–122. Springer Nature, Cham (2021)
29. Wang, W.Y.: "liar, liar pants on fire": A new benchmark dataset for fake news detection. CoRR **abs/1705.00648** (2017), http://arxiv.org/abs/1705.00648
30. Xie, Q., Dai, Z., Hovy, E.H., Luong, M., Le, Q.V.: Unsupervised data augmentation. CoRR **abs/1904.12848** (2019), http://arxiv.org/abs/1904.12848

.

B.3 A Fusion Approach for Enhanced Remote Sensing Image Classification

Vian Abdulmajeed Ahmad, Khaled Jouini, Amel Tuama & Ouajdi Korbaa
Proceedings of the 19^{th} International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - (Volume 2). Février 2024.

Lien : https://doi.org/10.5220/0012376600003660

CORE Rank : B (lors de la soumission)

ICORE Conference Portal

Back to search

Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (GRAPP and VISAPP combined from 2008)

Acronym: VISIGRAPP

DBLP Source: https://dblp.uni-trier.de/db/conf/visigrapp

Source: CORE2023

Rank: Multiconference

Field Of Research: 4603 - Computer vision and multimedia computation (h-index) (citation)

This conference was on the review list to retain current rank. Requested to retain B, decision Multiconference. (Data 1) (Decision)

Source: CORE2021

Rank: B

Field Of Research: 4603 - Computer vision and multimedia computation

Source: CORE2020

Rank: B

Field Of Research: 4603 - Computer vision and multimedia computation

Source: CORE2018

Rank: B

Field Of Research: 0801 - Artificial Intelligence and Image Processing†

Source: CORE2017

Rank: B

Field Of Research: 0801 - Artificial Intelligence and Image Processing†

A Fusion Approach for Enhanced Remote Sensing Image Classification

Vian Abdulmajeed Ahmed [1][a], Khaled Jouini [1][b], Amel Tuama [c] and Ouajdi Korbaa [1][d]
[1] *University of Sousse, MARS Research Lab, LR17ES05, ISITCom, 4011 H. Sousse, Tunisia*
[2] *Northern Technical University, Computer Engineering Techniques Department, Iraq*
vian.ahmad85@gmail.com, amel.tuama@ntu.edu.iq j.khaled@gmail.com,ouajdi.korbaa@centraliens-lille.org

Keywords: Remote Sensing, Land Cover Mapping, Features Fusion, Convolutional Neural Networks (CNNs), Scale-Invariant Feature Transform (SIFT), Image Classification.

Abstract: Satellite imagery provides a unique and comprehensive view of the Earth's surface, enabling global-scale land cover mapping and environmental monitoring. Despite substantial advancements, satellite imagery analysis remains a highly challenging task due to intrinsic and extrinsic factors, including data volume and variability, atmospheric conditions, sensor characteristics and complex land cover patterns. Early methods in remote sensing image classification leaned on human-engineered descriptors, typified by the widely used Scale-Invariant Feature Transform (SIFT). SIFT and similar approaches had inherent limitations in directly representing entire scenes, driving the use of encoding techniques like the Bag-of-Visual-Words (BoVW). While these encoding methods offer simplicity and efficiency, they are constrained in their representation capabilities. The rise of deep learning, fuelled by abundant data and computing power, revolutionized satellite image analysis, with Convolutional Neural Networks (CNNs) emerging as highly effective tools. Nevertheless, CNNs' extensive need for annotated data limits their scope of application. In this work we investigate the fusion of two distinctive feature extraction methodologies, namely SIFT and CNN, within the framework of Support Vector Machines (SVM). This fusion approach seeks to harness the unique advantages of each feature extraction method while mitigating their individual limitations. SIFT excels at capturing local features critical for identifying specific image characteristics, whereas CNNs enrich representations with global context, spatial relationships and hierarchical features. The integration of SIFT and CNN features helps thus in enhancing resilience to perturbations and generalization across diverse landscapes. An additional advantage is the adaptability of this approach to scenarios with limited labelled data. Experiments on the EuroSAT dataset demonstrate that the proposed fusion approach outperforms SIFT-based and CNN-based models used separately and that it achieves either better or comparable results when compared to existing notable approaches in remote sensing image classification.

1 INTRODUCTION

Satellite imagery plays a pivotal role in various applications, including land cover mapping, environmental monitoring, disaster assessment, and urban planning. Thanks to advances in Earth observation technology, the volume of remote sensing images is rapidly increasing. Understanding these vast and complex images has become an increasingly important and challenging task (Janga et al., 2023). At the heart of this challenge lies scene and images classification, a complex task that has garnered significant attention from the research community. The central goal of remote sensing scene classification is to accurately assign predefined semantic categories to images. Scene classification requires a high degree of accuracy and adaptability, as the scenes encountered in practice are typically diverse, spanning both rural and urban

[a] https://orcid.org/0009-0002-5924-6139
[b] https://orcid.org/0000-0002-3802-9074
[d] https://orcid.org/0000-0003-4462-1805

landscapes(Wang et al., 2022). To meet the requirements of these diverse applications, a classification model must be equipped to handle variations in scale, atmospheric conditions, and noise, while also being capable of recognizing complex spatial patterns(Weiss et al., 2020).

In the early stages of remote sensing scene classification, many methods relied heavily on human-crafted features, with Scale-Invariant Feature Transform (SIFT) being a prominent example. SIFT and similar methods faced the challenge of directly representing an entire image due to their inherent local nature (Weinzaepfel et al., 2011). To address this limitation, local descriptors often employed encoding methods, such as the popular Bag-of-Visual-Words (BoVW). While these encoding methods offered simplicity and efficiency, they simultaneously had limited representation capabilities, especially as they do not allow to represent spatial relationships (Cheng et al., 2019). In response to these limitations, unsupervised learning methods, which autonomously learn features from unlabeled images, emerged as an attractive alternative to human-crafted descriptors. These methods, often employing techniques like k-means clustering, presented a promising avenue for scene classification. Nevertheless, unsupervised methods lack the supervised learning's advantage of having class labels to guide the feature learning process, which often lead to learn features that are not relevant to the classification task (Cheng et al., 2019).

The advances in deep learning theory, coupled with the increased availability of remote sensing data and parallel computing resources, ushered in a new era for remote sensing image scene classification. Deep learning models have demonstrated their prowess in feature description across various domains, and remote sensing image scene classification was no exception (Aksoy et al., 2023). Convolutional Neural Networks (CNNs) emerged as a powerful tool, pushing the boundaries of classification accuracy in the field. However, the data-hungry nature of CNNs and their extensive need for annotated training data limit their scope of application.

In this study, we investigate the fusion of two distinctive feature extraction methods, namely SIFT and CNN, within the framework of Support Vector Machines (SVM) for remote sensing images classification. This approach aims to harness the benefits of both feature extraction approaches, while overcoming the limitations of each method used separately. SIFT excels in capturing unique local features that are essential for recognizing unique characteristics within an image. The global context and the hierarchical features learned by CNNs contribute to better generalization, ensuring that the model can accurately classify scenes exhibiting complex patterns that are challenging to capture with local features alone. An additional advantage of this approach is its adaptability to situations with limited labeled data, a common issue in remote sensing.

The EuroSAT dataset (Cheng et al., 2020) is a widely recognized and extensively employed collection of satellite images containing 10 classes of land cover. It consists of 27,000 images collected by the Sentinel-2 satellite, having a spatial resolution of 10 meters. Experiments on EuroSAT dataset demonstrate that our fusion approach outperforms, not only SIFT and CNN used separately, but also existing remote sensing image classification approaches.

The remainder of this paper is organized as follows. Section II briefly reviews related work. Section III presents our features fusion approach. Section IV provides a comparative experimental study on EuroSAT dataset. Finally, section V concludes the paper.

2 RELATED WORK

Land Use and Land Cover (LULC) classification has garnered substantial attention within the scientific community, with numerous studies and reviews dedicated to the comparison of various approaches and emerging trends. For the sake of conciseness and due to lack of space, we mainly focus in the sequel on approaches that employ the EuroSAT dataset used in our experimental study or presenting similarities with our approach. Existing approaches and studies can be broadly classified into two families: Machine Learning (ML)-based algorithms and Deep Learning (DL)-based methods(Yaloveha et al., 2023).

The studies presented in (Hu et al., 2014), (Chen & Tian, 2015), and (Thakur & Panse, 2022) are representative of ML-based approaches. Hu et al. (2014) proposed a method that utilizes randomly sampled image patches for Unsupervised Feature Learning (UFL) in image classification. They applied the BOVW model to this approach and conducted experiments on an aerial scene dataset. The experiments on the dataset present encouraging results with an accuracy of 90.03%. (Chen & Tian, 2015) introduced the Pyramid of Spatial Relations (PSR) model, designed to incorporate both relative and complete spatial information into the BOVW framework for LULC classification. Experiments conducted on a high-resolution remote sensing image revealed that the PSR model achieves an average

classification accuracy of 89.1%. In (Thakur & Panse, 2022), the authors evaluate the performance of four machine learning algorithms: decision tree (DT), k-nearest neighbor (KNN), support vector machine (SVM), and random forest (RF). The results indicate that RF exhibits superior performance compared to DT, KNN, and SVM, while SVM and DT exhibit similar levels of effectiveness.

The studies (P Helber et al., 2018), (Dewangkoro & Arymurthy, 2021) and (Temenos et al., 2023) are representative of DL-based studies. The authors in (Temenos et al., 2023) introduce an interpretable deep learning framework for LULC classification using SHapley Additive exPlanations (SHAPs). (Temenos et al., 2023) uses a compact CNN model for images classification and then feeds the results to a SHAP deep explainer. Experimental results on the EuroSAT dataset demonstrate the CNN's accurate classification with an overall accuracy of 94.72%, whereas the classification accuracy on three-band combinations on each of the dataset's classes highlight its improvement when compared to standard approaches. The SHAP explainable results of the proposed framework shield the network's predictions by showing correlation values that are relevant to the predicted class, thereby improving the classifications occurring in urban and rural areas with different land uses in the same scene.

Another interesting DL-based study is presented in (Dewangkoro & Arymurthy, 2021). The approach of (Dewangkoro & Arymurthy, 2021) uses different CNN architectures for feature extraction, namely VGG19, ResNet50, and InceptionV3. Then, the extracted feature is recalibrated using Channel Squeeze & Spatial Excitation (sSE) block. The approach also uses SVM and Twin SVM (TWSVM) as classifiers. VGG19 with sSE block and TWSVM achieved the highest experimental results with 94.57% accuracy, 94.40% precision, 94.40% recall, and 94.39% F1-score.

In the study by (P Helber et al., 2018), the authors compares vaious CNN architectures, namelya shallow CNN, a ResNet50-based model model and a GoogleNet-based model. The overall classification accuracy achieved is 89.03%, 98.57%, 98.18% respectively. The authors also evaluated the performance of the Bagof-Visual-Words (BoVW) approach using SIFT features and a trained SVM. The study of (P Helber et al., 2018) shos that all CNN approaches outperform the BoVW method and, overall, deep CNNs perform better than shallow CNNs which achieves an overall accuracy of 89.03% on EuroSAT dataset.

The aforementioned studies demonstrate that there is no one-size-fits-all algorithm that can attain the highest accuracy for all the classes under consideration. Furthermore, as this section highlights, existing approaches tend to concentrate on either classical machine learning methods or deep learning algorithms, with none delving into the advantages that can be derived from integrating classical methods with deep learning algorithms. Our study aims to underscore and quantify the potential benefits of such an integration.

3 FEATURES FUSION VS. HAND-CRAFTED AND CNN-LEARNED FEATURES

Hand-crafted features have played a significant role in computer vision applications, particularly image classification. These features are derived through non-learning processes, directly applying various operations to image pixels. They offer advantages like rotation and scale invariance, achieved by efficiently encoding local gradient information. However, hand-crafted features have three notable limitations (Tsourounis et al., 2022): (*i*) They provide a low-level representation of data and lack the ability to offer an abstract representation crucial for recognition tasks; (*ii*) Local descriptors like SIFT do not yield a fixed-length vector representation of input images, necessitating additional logic for local descriptor encoding, such as Bag-of-Visual-Words (BoVW); and (*iii*) Their capacity is fixed and limited by a predefined mapping from data to feature space, regardless of specific recognition needs.

In the past decade, hand-crafted methods have been largely supplanted by deep Convolutional Neural Networks (CNNs). CNNs employ an end-to-end learning approach, typically in a supervised manner. Each input image is associated with a ground-truth label, and the CNN model's weights are updated iteratively until the model's output aligns with the label. This way, CNNs construct hierarchical feature representations through a learning process that minimizes a defined cost function. CNNs learn feature representation and encoding directly from images, resulting in a model that provides high-level feature representations once trained on a particular dataset and task. However, CNNs demand extensive data and are sensitive to data quality, making them dependent on large annotated datasets while posing challenges related to achieving scale, rotation, or geometric invariance.

In this study, we investigate the synergy between local descriptors (SIFT) and CNN-learned descriptors. To this end, we compare the fusion of CNN-SIFT features to the cases where SIFT and CNN are used separately. The framework of the proposed models is shown in Figure1. It is worth noting that to gain a more comprehensive understanding of the isolated impact of the fusion approach without any additional considerations or optimizations, we opted for a basic SIFT-based model and a straightforward CNN architecture for feature extraction. While we acknowledge that more complex CNN architectures, such as those used in (Patrick Helber et al., 2019) and (Dewangkoro & Arymurthy, 2021), have the potential to further improve predictive performance, such complex architectures make it difficult to quantify the specific advantages gained from incorporating SIFT-based descriptors. The three models that we study are presented in the sequel.

3.1 Model 1: CNN-based Remote Sensing Image Classification

The architectural components of the explored CNN model are depicted in Figure 2. The model incorporates two convolutional layers to capture essential image features. The initial convolutional layer operates on input images with dimensions of (64, 64, 3) and employs 32 filters with Rectified Linear Unit (ReLU) activation functions, each having a size of 3x3. This layer effectively extracts fundamental characteristics from the input data. The output features from the first layer are further refined by a second convolutional layer, consisting of 64 filters, each with a size of 3x3. The model integrates two max-pooling layers for spatial dimension reduction. The first max-pooling layer reduces the spatial dimensions of the feature maps by a factor of two, enhancing computational efficiency. The second max-pooling layer further compresses the spatial dimensions, facilitating more abstract feature extraction. A flattened layer precedes the fully connected layers, transforming the 2D feature maps into a 1D vector. The network architecture comprises also two dense layers, with the first layer housing 128 neurons activated by ReLU. This configuration allows the model to learn intricate representations from the data. For multi-class classification tasks, the final layer encompasses ten neurons, utilizing the SoftMax activation function to generate class probabilities. During training, we employ the Adam optimizer and a sparse categorical cross-entropy loss function to optimize the model. The primary objective is to minimize the loss and ensure accurate categorization through the training process. "Rather than training the investigated CNN model from scratch, we employ transfer learning and use a pre-trained model with weights acquired from the ImageNet dataset (Abou Baker et al., 2022) .

3.2 Model 2: SIFT-based Remote Sensing Image Classification

The different steps of the second studied approach are illustrated in Figure 3. The first step involves the conversion of the original satellite images into grayscale format, simplifying the data while retaining essential visual characteristics. Following this conversion, the SIFT algorithm is applied to identify key points and extract local feature descriptors. These SIFT descriptors represent distinctive image regions and are crucial for capturing unique visual patterns. To further streamline the feature representation and enable efficient classification, we adopt the Bag-of-Visual-Words (BoVW). Here, the extracted SIFT descriptors are quantized into visual words, reducing the feature dimensionality and forming the basis for image representation. The quantization process is facilitated by k-means clustering, which groups similar descriptors into clusters, and the cluster centers become the visual words. Finally, we employ a Support Vector Machine (SVM) model to train on the BoVW-represented satellite images.

3.3 Model 3: Fusion of SIFT and CNN Features

This paper introduces a novel hybrid model that synergizes the strengths of CNN-learned features with SIFT descriptors to enhance remote sensing image classification (Figure 4). The proposed approach harnesses the power of the CNN to extract high-level, semantically rich features, providing a global understanding of the image. It also employs the SIFT detector (Open CV SIFT) to capture fine-grained, local details, benefiting from its robustness to various transformations (*e.g.* scale and rotation).
The SIFT features are flattened and have lengths that are either zero-padded or truncated to 128. To extract deep features, we leverage a pre-trained CNN architecture with weights sourced from ImageNet. The base model is modified by excluding the final fully connected layers (*i.e. Include top=False*), retaining only the convolutional layers. The input images, which are initially of varying dimensions, are pre-processed and resized to meet the (224, 224, 3) input shape requirement of the CNN model.

Once the SIFT and the CNN features are generated, a unified feature vector is produced by horizontally stacking the CNN features with the truncated/flattened SIFT features. Each image is represented by this feature vector combination, which is a rich representation, encapsulating both global and local information. For the task of classification, we employ a straightforward Support Vector Machine (SVM) with a radial basis function (RBF) kernel. The RBF kernel's flexibility enables the model to capture complex decision boundaries in the feature space. As demonstrated in our experimental study, this synergy between deep learning-based features from the CNN a conventional computer vision characteristic from SIFT yields enhanced classification performance.

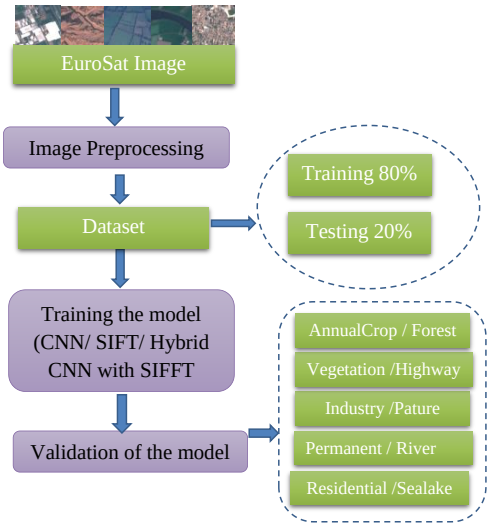


Figure 1 Frame Work for the proposed model

Input (64x64x3 Image Data)
Conv2D (32) 3x3,ReLU
MaxPooling 2x2
Conv2D (64) 3x3, ReLU
MaxPooling 2x2
Flatten
Dense (128) | ReLU
Dense (10) Softmax
Output (10 classes)

Figure 2 Proposed CNN architecture for classification Remote sensing images

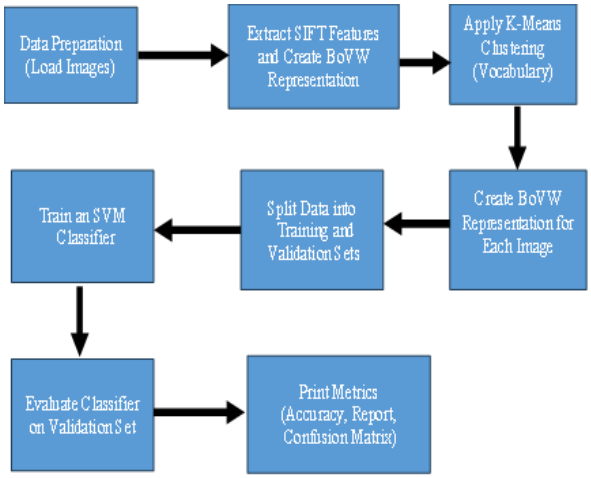


Figure 3 Proposed SIFT procedure for classification Remote Sensing Images

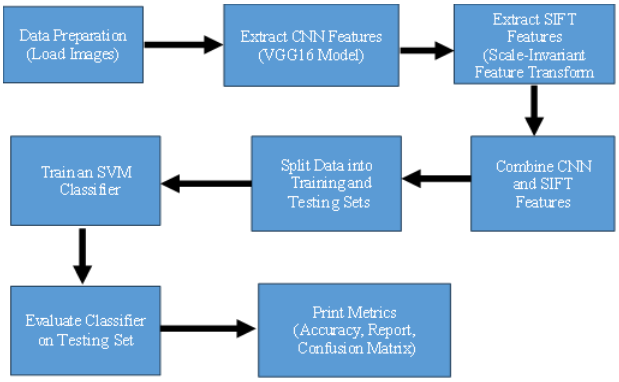


Figure 4 Proposed Hybrid CNN with SIFT models

4 EXPERIMENTAL STUDY

The EuroSAT dataset (Patrick Helber et al., 2019) used in our experiments, is openly and freely provided by the Copernicus Earth observation program. The dataset is generated with 27,000 labeled and georeferenced image patches, where the size of each image patch is 64_64 m. To ech of the 10 classes of the dataset corresponds 2000 to 3000 images. The LULC classes in this dataset are *permanent crop*, *annual crop*, *pastures*, *river*, *sea & lake*, *forest*, *herbaceous vegetation*, *industrial building*, *highway* and *residential building* (Patrick Helber et al., 2019) .

Figure 5 Sample images of Eurosat dataset (Helber et al., 2019)

In our experiments, 80% of the dataset was allocated for training, while the remaining instances were reserved for testing. The studied CNN architecture was implemented using Tensorflow (Yu et al., 2019) and Keras (Lee & Song, 2019). OpenCV (Culjak et al., 2012) was used for SIFT features generation. All methods were run using their default settings, and no special tuning was done.

The primary goal of our study is to bring to light the advantages that can be drawn from combining SIFT and CNN features. We then compare and contrast three different remote sensing classification models: CNN-based, SIFT-based, and a fusion approach combining SIFT and CNN features. Figures 6, 7, 8, 9, 10 and 11 illustrate, respectively, the confusion matrix and the detailed classification report of the CNN-based model (Model 1), The SIFT-based model (Model 2) and the model based on a fusion of CNN and SIFT features (Model 3). Table 4.1. provides the overall accuracy of the different models. As shown in these figures and in Table 4.1, the features fusion approach by far outperforms the SIFT-based model and the CNN-based model, allowing an enhancement of accuracy of 64.29% and 84.10% respectively.

Table 4.2 compares the accuracy results of our models with some of notable existing approaches. As shown in Table 4.2 our approach achieves an overall accuracy of 92%, and, except for the approach presented in (Dewangkoro & Arymurthy, 2021), it outperforms all other models. The "BoVW (SVM, SIFT, k = 500)" model, based on BoVW with SVM and SIFT, achieves an overall accuracy of 70%. The "UFL" model achieves an overall accuracy of 90%, demonstrating the effectiveness of unsupervised feature learning. The "Pyramid of Spatial Relations"(Chen & Tian, 2015) model reaches 89% accuracy, emphasizing the importance of capturing spatial relationships. The approach presented in (Dewangkoro & Arymurthy, 2021) uses different CNN architectures for feature extraction, namely VGG19, ResNet50, and InceptionV3, and achieves an accuracy of 94%. The approach (Dewangkoro & Arymurthy, 2021) inherits the advantages and limitations of deep neural architectures. Our approach achieves comparable performance to that of (Dewangkoro & Arymurthy, 2021) while being less resource-intensive and less reliant on the availability of massive labelled data.

As mentioned earlier, in order to better understand the impact of the fusion approach in isolation, we implemented a basic SIFT-based model and a simple CNN architecture for feature extraction. However, it is worth noting the use in our approach of more sophisticated CNN architectures, such as those used in (Helber et al., 2019) and (Dewangkoro & Arymurthy, 2021), have the potential to further enhance predictive performance.

```
[[526   2  16  13   0  15  16   0   4   3]
 [  0 577   2   0   0  21   0   0   1   5]
 [  6   3 479  16   3   8  71  10   5   1]
 [ 16   0  23 358   6  11  42  14  45   0]
 [  0   0  10  41 396   0   7  38   1   0]
 [ 15   5  20  13   0 350   8   0   9   3]
 [ 23   0  54  33   1   9 345   2   6   0]
 [  0   0  14   9   3   0   6 565   0   0]
 [ 37  12  13  84   1  13   6   3 320   1]
 [  4   4   1   1   0   3   0   0   7 586]]
```

Figure 6 Confusion Matrix for CNN model for remote sensing images classification

	precision	recall	f1-score	support
0	0.84	0.88	0.86	595
1	0.96	0.95	0.95	606
2	0.76	0.80	0.78	602
3	0.63	0.70	0.66	515
4	0.97	0.80	0.88	493
5	0.81	0.83	0.82	423
6	0.69	0.73	0.71	473
7	0.89	0.95	0.92	597
8	0.80	0.65	0.72	490
9	0.98	0.97	0.97	606

Figure 7 Precision, Recall, F1-Score for CNN model for remote sensing images classification

	precision	recall	f1-score	support
AnnualCrop	0.65	0.71	0.68	520
Forest	0.00	0.00	0.00	66
HerbaceousVegetation	0.40	0.36	0.38	461
Highway	0.53	0.45	0.49	483
Industrial	0.74	0.86	0.79	527
Pasture	0.28	0.65	0.40	244
PermanentCrop	0.51	0.41	0.45	483
Residential	0.77	0.78	0.78	592
River	0.42	0.29	0.35	458
SeaLake	0.00	0.00	0.00	54

Figure 8 Precision, Recall, F1-Score for SIFT model for remote sensing images classification

```
[[371   0  20  15   0  67  12   0  35   0]
 [  2   0   5   0   0  57   0   0   2   0]
 [ 21   1 167  29  39 100  51  24  29   0]
 [ 46   0  36 216  25  20  56  23  61   0]
 [  2   0   3  15 454   0  21  30   2   0]
 [ 25   0  28   2   0 159   6  13  11   0]
 [ 30   0  59  39  69  35 198  28  25   0]
 [  2   0  34  17  27  24  14 464  10   0]
 [ 66   0  57  73   3  73  34  17 135   0]
 [  7   0   7   1   0  26   0   1  12   0]]
```

Figure 9 Confusion Matrix for SIFT model for remote sensing images classification

	precision	recall	f1-score	support
AnnualCrop	0.91	0.97	0.94	534
Forest	0.88	0.93	0.90	69
HerbaceousVegetation	0.89	0.90	0.89	473
Highway	0.90	0.87	0.89	479
Industrial	0.93	0.96	0.94	512
Pasture	0.93	0.90	0.91	251
PermanentCrop	0.90	0.87	0.88	486
Residential	0.97	0.97	0.97	591
River	0.91	0.88	0.89	441
SeaLake	0.96	0.88	0.92	52

Figure 10 Precision, Recall, F1-Score for Hybrid CNN with SIFT model for remote sensing images classification

```
[[516   1   4   1   0   2   5   0   4   1]
 [  0  64   1   0   0   3   0   0   1   0]
 [  2   2 425   7   5   0  20   7   5   0]
 [ 12   2   2 418   5   4   9   2  24   1]
 [  0   0   1   6 489   0   7   9   0   0]
 [  5   3  11   0   0 225   4   0   3   0]
 [ 16   0  28   6  11   1 422   1   1   0]
 [  0   1   3   0  12   0   1 574   0   0]
 [ 13   0   3  27   1   7   2   0 388   0]
 [  3   0   0   0   0   1   0   0   2  46]]
```

Figure 11 Confusion Matrix for Hybrid CNN with SIFT model for remote sensing images classification

Table 4.1 The results of accuracy for the proposed models

Model	Accuracy
CNN	0.83
SIFT	0.56
Fusion of CNNs and SIFT features	**0.92**

Table 4.2 The accuracy Result of the proposed model compare with related work

Models	Accuracy
CNN two layer (Helber et al., 2018)	0.87
BoVW (SVM, SIFT, k = 500) (Helber et al., 2018)	0.70
UFL (Hu et al., 2014)	0.90
Pyramid of spatial relatons(Chen & Tian, 2015)	0.89
Combination of different CNNs deep architectures (Dewangkoro & Arymurthy, 2021)	0.94
Fusion of CNNs and SIFT features (proposed model)	**0.92**

5 CONCLUSION AND FUTURE WORK

5.1 Conclusion

Remote sensing image classification is a crucial task for various critical applications, including land cover mapping, environmental monitoring, and disaster response. In this work we investigated the fusion of hand-crafted features (SIFT) and CNN-learned features to enhance remote sensing image classification. SIFT excels in capturing local features essential for discerning specific image attributes. Meanwhile, CNNs' ability to learn global context and hierarchical features, enhances generalization and allows accurate classification of scenes with complex patterns. The experimental study conducted over the EuroSAT dataset shows that our fusion approach allows a substantial classification enhancement with regards to CNN and SIFT used separately: up to 10.84% accuracy enhancement when compared to CNN and up to 64.29% enhancement when compared to SIFT. Although our fusion approach was implemented using straightforward SIFT-based Model and CNN architecture (to better isolate the benefits of features fusion), our experimental study shows that it achieves better or comparable results with notable existing remote sensing image classification approaches.

5.2 Future work

The promising obtained results pave the way for the exploration of other applications and further forms of collaboration between classical hand-crafted features and modern deep features. We are currently exploring two research directions. The first involves remote sensing images enhancement during registration, which aims at improving the quality of remote sensing images to make them more amenable to subsequent analysis. The second focuses on the detection of changes in images captured within the same geographic areas but at different time points. Such change detection is crucial several critical in domains such as environmental monitoring. To this end we are currently investigating the integration of SIFT and Siamese networks for efficient change detection in remote sensing image.

REFERENCES

Abou Baker, N., Zengeler, N., & Handmann, U. (2022). A Transfer Learning Evaluation of Deep Neural Networks for Image Classification. In *Machine Learning and Knowledge Extraction* (Vol. 4, Issue 1,pp. 22–41). https://doi.org/10.3390/make4010002

Aksoy, M. Ç., Sirmacek, B., & Ünsalan, C. (2023). Land classification in satellite images by injecting traditional features to CNN models. *Remote Sensing Letters*, *14*(2), 157–167. https://doi.org/10.1080/2150704X.2023.2167057

Chen, S., & Tian, Y. (2015). Pyramid of Spatial Relatons for Scene-Level Land Use Classification. *IEEE Transactions on Geoscience and Remote Sensing*, *53*(4), 1947–1957. https://doi.org/10.1109/TGRS.2014.2351395

Cheng, G, Xie, X., Han, J., Guo, L., & Xia, G.-S. (2020). Remote Sensing Image Scene Classification Meets Deep Learning: Challenges, Methods, Benchmarks, and Opportunities. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *13*, 3735–3756. https://doi.org/10.1109/JSTARS.2020.3005403

Cheng, Gong, Xie, X., Han, J., Guo, L., & Xia, G. S. (2019). Remote Sensing Image Scene Classification Meets Deep Learning: Challenges, Methods, Benchmarks, and Opportunities. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *13*(X), 37353756. https://doi.org/10.1109/JSTARS.2020.3005403

Culjak, I., Abram, D., Pribanic, T., Dzapo, H., & Cifrek, M. (2012). A brief introduction to OpenCV. *2012 Proceedings of the 35th International Convention MIPRO*, 1725–1730.

Dewangkoro, H. I., & Arymurthy, A. M. (2021). Land use and land cover classification using CNN, SVM, and Channel Squeeze & Spatial Excitation block. *IOP Conference Series: Earth and Environmental Science*, *704*(1). https://doi.org/10.1088/1755-1315/704/1/012048

Helber, P, Bischke, B., Dengel, A., & Borth, D. (2018). Introducing Eurosat: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification. *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, 204–207. https://doi.org/10.1109/IGARSS.2018.8519248

Helber, Patrick, Bischke, B., Dengel, A., & Borth, D. (2019). Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *12*(7), 2217–2226. https://doi.org/10.1109/JSTARS.2019.2918242

Hu, F., Xia, G.-S., Wang, Z., Zhang, L., & Sun, H. (2014). Unsupervised feature coding on local patch manifold for satellite image scene classification. *2014 IEEE Geoscience and Remote Sensing Symposium*, 1273–1276. https://doi.org/10.1109/IGARSS.2014.6946665

Janga, B., Asamani, G. P., Sun, Z., & Cristea, N. (2023). A Review of Practical AI for Remote Sensing in Earth Sciences. In *Remote Sensing* (Vol. 15, Issue 16). https://doi.org/10.3390/rs15164112

Lee, H., & Song, J. (2019). Introduction to convolutional neural network using Keras; an understanding from a statistician. *Communications for Statistical Applications and Methods*, *26*(6), 591–610.

Temenos, A., Temenos, N., Kaselimi, M., Doulamis, A., & Doulamis, N. (2023). Interpretable Deep Learning Framework for Land Use and Land Cover Classification in Remote Sensing Using SHAP. *IEEE Geoscience and Remote Sensing Letters*, *20*, 1–5. https://doi.org/10.1109/LGRS.2023.3251652

Thakur, R., & Panse, P. (2022). Classification Performance of Land Use from Multispectral Remote Sensing Images using Decision Tree, K-Nearest Neighbor, Random Forest and Support Vector Machine Using EuroSAT. *nternational Journal of Intelligent Systems and Applications in Engineering IJISAE*, *2022*(1s),67–77. https://github.com/phelber/EuroSAT

Tsourounis, D., Kastaniotis, D., Theoharatos, C., Kazantzidis, A., & Economou, G. (2022). SIFT-CNN: When Convolutional Neural Networks Meet Dense SIFT Descriptors for Image and Sequence Classification. *Journal of Imaging*, *8*(10). https://doi.org/10.3390/jimaging8100256

Wang, X., Xu, H., Yuan, L., Dai, W., & Wen, X. (2022). A Remote-Sensing Scene-Image Classification Method Based on Deep Multiple-Instance Learning with a Residual Dense Attention ConvNet. In *Remote Sensing* (Vol. 14, Issue 20). https://doi.org/10.3390/rs14205095

Weinzaepfel, P., Jégou, H., & Pérez, P. (2011). Reconstructing an image from its local descriptors. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 337–344. https://doi.org/10.1109/CVPR.2011.5995616

Weiss, M., Jacob, F., & Duveiller, G. (2020). Remote sensing for agricultural applications: A meta-review. *Remote Sensing of Environment*, *236*, 111402. https://doi.org/https://doi.org/10.1016/j.rse.2019.111402

Yaloveha, V., Podorozhniak, A., Kuchuk, H., & Garashchuk, N. (2023). Performance Comparison of Cnns on High-Resolution Multispectral Dataset Applied To Land Cover Classification Problem. *Radioelectronic and Computer Systems*, *2023*(2(106)), 107–118. https://doi.org/10.32620/REKS.2023.2.09

Yu, L., Li, B., & Jiao, B. (2019). Research and Implementation of CNN Based on TensorFlow. *IOP Conference Series: Materials Science and Engineering*, *490*, 42022. https://doi.org/10.1088/1757-899X/490/4/042022

.

C.1 Drift-Driven Regression for Predicting the Evolution of Pandemics

Khaled Jouini & Ouajdi Korbaa

IADIS International Conference Applied Computing (IADIS AC). 2023.
pp. 77-84, ISBN : 978-989-8704-53-5

Lien : https://www.iadisportal.org/digital-library/drift-driven-regression-for-predicting-the-evolution-of-pandemics

CORE Rank : C

ICORE Conference Portal

Back to search

IADIS International Conference Applied Computing

Acronym: IADIS AC

DBLP Source: N/A

Source: CORE2023

Rank: C

Field Of Research: 4601 - Applied computing

Source: CORE2021

Rank: C

Field Of Research: 4601 - Applied computing

Source: CORE2020

Rank: C

Field Of Research: 4601 - Applied computing

Source: CORE2018

Rank: C

Source: CORE2017

Rank: C

Source: CORE2014

Rank: C

Source: CORE2013

Rank: C

DRIFT-DRIVEN REGRESSION FOR PREDICTING THE EVOLUTION OF PANDEMICS

Khaled Jouini and Ouajdi Korbaa
University of Sousse, MARS Research Lab, LR17ES05, ISITCom
4011 H. Sousse, Tunisia

ABSTRACT

Pandemics will never cease to emerge and threaten public health and the global economy. Predicting the evolution of a pandemic is of paramount interest as it enables policymakers understand the potential spread of a virus and make informed decisions to mitigate its impact. *Concept drifts* refer to situations where the relationship between the input features (*model input*) and the learning targets (*model output*) changes or evolves over time. Concept drifts are common in pandemic curves, not only because new variants appear over time, but also due to factors such as seasonality, policy responses to the pandemic, and changes in the way the disease is treated. Without proper intervention, the accuracy of conventional (*batch*) machine learning models will deteriorate after drift occurs, since they were trained on outdated data. *Incremental learning* is an alternative to batch learning, where the training phase never ends, and the model is incrementally updated as new data becomes available. While incremental models are in general less accurate than batch models, they adapt better to concept drifts as they are continuously refined using the most recent data. Batch and incremental learning are often considered as two distinct and mutually exclusive approaches (Montiel et al., 2018a). In this work we propose CDR (*Collaborative Drift- Driven Regression*), a novel collaborative regression strategy where incremental and batch regressors work together to complement each other's strengths, ergo the overall predictive performance. Experiments conducted on COVID-19 pandemic data, show that CDR is an efficient collaborative learning strategy that yields better results than the underlying batch and incremental models used separately.

KEYWORDS

Concept Drift, Incremental Learning, Collaborative Learning, Pandemic Forecasting

1. INTRODUCTION

The risk of extreme pandemics like COVID-19 is increasing due to climate change, ease of global travel, and increasing rates of disease emergence from animal reservoirs (Marani et al., 2021). The accurate prediction of a pandemic's evolution is crucial for improved preparedness and prompt responses. It is however a particularly challenging task due to factors such as, changes in the way the virus spreads or mutates, changes in testing or reporting practices, and changes in the way the disease is treated. In machine learning and related fields, changes over time in the relationship between input data and the learning target are known as *Concept Drifts* (*a.k.a. data non-stationarity*) (Bifet & Gavaldà, 2007).

Most state-of-the-art machine learning algorithms, referred to as *batch learners* in the sequel, operate under the premise of data stationary, and assume that all training data is available prior to the learning process. Without proper intervention, the predictive performance of batch learners inevitably deteriorates as drifts occur, since they were trained on historical data and no longer accurately reflect the current relationship between input and target variables. A common intervention to handle drifts is to periodically *retrain* the batch model to take into account recent observations. Besides the computational burden, model retraining raises two significant challenges: (*i*) determining when a model is no longer valid and requires retraining (*i.e.* stability-plasticity dilemma); and (*ii*) deciding how much new data to collect, given that collecting more data enhances the chances of generating an accurate model, but also delays the replacement of the old poorly performing model.

Incremental learning (*a.k.a. online learning* or *lifelong learning*) is an alternative to batch learning, wherein the model is trained on small amounts of data at a time, rather than in a single batch. In incremental learning, the training phase is perpetual, and the model undergoes continuous and incremental refinement as

new data becomes available. This continual updating enables incremental models to exhibit superior adaptability to drifts when compared to batch models. Another desirable feature of incremental learning is its *"anytime property"* which refers to the ability of a model to make predictions at any point in time during the learning process. By being responsive to changes, easy to maintain (models do not need to be retrained), and able to start making predictions after the first few training instances, incremental learning seems to be well-suited to the requirements of modeling the evolution of a pandemic. The downside of incremental learning, however, is that it builds models by making assumptions on upcoming data (an incremental model is in fact an approximation of the corresponding batch model) (Bifet & Gavaldà, 2007). In contrast to incremental algorithms, batch algorithms need time to collect enough data before building a model, but once the model is built, it is often more accurate than the corresponding incremental model.

Batch and incremental learning are commonly considered as distinct and mutually exclusive approaches (Montiel et al., 2018a). In this paper, we propose CDR (*Collaborative Drift-Driven Regression*), a novel collaborative regression strategy in which incremental and batch regressors work together to complement their strengths, with the ultimate goal of accurately predicting the evolution of pandemics. CDR continuously refines the underlying incremental model as data arrives, retrains a new batch model on recent observations whenever it detects a drift and dynamically selects the best performing model to produce predictions. Experiments conducted on COVID-19 data show that CDR yields better results compared to using the underlying incremental and batch regressors separately.

The remainder of this paper is organized as follows. Section 2 briefly reviews the main concepts related to incremental learning and provides an overview of related work. Section 3 details our CDR approach. Section 4 outlines the experimental evaluation and examines the main findings. Conclusions and future work are discussed in Section 5.

2. PRELIMINARIES AND RELATED WORK

In sensitive areas like pandemics forecasting, interpretability matters as much as accurate predictions, and white-box models are often preferred to black-box models (Salah, I. et al., 2023). Decision Trees are among the most popular white-box models. This section first introduces the key concepts related to incremental trees and then briefly reviews the main approaches utilizing them for pandemics forecasting.

2.1 Incremental Trees

A decision tree is learned top-down by recursively replacing leaves by test nodes. The recursion is completed when a node is deemed homogeneous enough, or when splitting no longer improves predictions. Batch trees scan the entire dataset to discover the attribute leading to the highest homogeneity. The aforementioned process cannot be adopted directly in contexts like pandemics forecasting where only a small fraction of data is accessible during learning. The *Hoeffding Tree* (HT) (Domingos & Hulten, 2000) is the de-facto standard in stream mining and has inspired many state-of-the-art incremental algorithms. The main idea behind HT is that a small fraction of data can often be enough to choose the best splitting attribute (*i.e.* the attribute leading to the highest homogeneity). This idea is supported by the *Hoeffding Bound* which states that, with probability $(1-\delta)$, the true mean of a random variable of range R will not differ from the estimated mean after n independent observations by more than (Domingos & Hulten, 2000):

$$\epsilon = \sqrt{\frac{R^2 \ln(1/\delta)}{2n}}$$

For the purpose of deciding which attribute to split on, the random variable being estimated is the difference in information gain between the best and second-best attributes, *resp.* referred to in the sequel as X_a and X_b. If the computed difference of information gains between X_a and X_b is higher than ϵ, the algorithm asserts with confidence $1-\delta$, that X_a will always remain a better split option than X_b. The *Fast Incremental Model Tree with Drift Detection* (FIMT-DD) (Ikonomovska et al., 2011) is among the state-of-the-art incremental regression trees. Like the *Hoeffding Adaptive Tree* (HAT) (Domingos & Hulten, 2000),

FIMT-DD is an adaptive incremental tree that encompasses local change detectors. The salient features of FIMT-DD are synthesized in the following points.

Splitting Criterion. FIMT-DD uses the *standard deviation reduction* (SDR) as splitting criterion. Similarly, to HT and HAT, FIMT-DD uses the Hoeffding Bound to control the risk that, as data arrives, the merit of splitting on X_b exceeds the merit of splitting on X_a. In practice, before splitting a node, FIMT-DD waits until the following condition is met.

$$\frac{SDR(X_b)}{SDR(X_a)} < 1 - \epsilon$$

Linear model at the leaves. FIMT-DD trains a perceptron at each leaf of the tree. The weights of the perceptrons are continuously updated as new data arrives using the incremental stochastic gradient descent and with the objective of minimizing the mean squared error. In addition to their proven effectiveness, perceptrons have the crucial advantage of naturally adapting to drifts (Ikonomovska et al., 2011).

Drift handling. FIMT-DD uses the Page-Hinkley (PH) change detection test (Mouss et al., 2004) at inner nodes to detect changes in the error rate. When a change is detected in a node *inner*, an alternate tree rooted at *inner* is grown with new incoming instances: every new instance that reaches *inner* is used for growing both subtrees. The new subtree replaces the original one when (and if) it performs better.

2.2 Related Work

The volume of literature on predicting the evolution of the COVID-19 pandemic is monumental and covers a wide range of techniques (Miralles-Pechuán et al., 2023). Despite the abundance of studies, little work has been dedicated to incremental learning (Miralles-Pechuán et al., 2023). Existing incremental approaches can be broadly classified into two families: *compartmental models* and *machine learning models*. Machine learning models rely on historical data to forecast future outcomes. Compartmental models divide the population into different groups and use differential equations to model the transition of individuals between these groups. The approach of (Camargo et al., 2022) is representative of compartmental methods and involves a two-component architecture, where the first component is a feature engineering process that selects the predictor variables used by the predictive models of the second component. Specifically, the first component analyzes the temporal dependencies between the SEIRD variables (*Susceptible, Exposed, Infected, Recovered, and Dead*) and identify, for each target variable, the best subset of predictors. The second component of the architecture employs an ensemble learning approach, where various models are trained using different batch and incremental algorithms. The final output is chosen based on the predictions of the top-performing model. When the predictions of the top-performing model are not deemed accurate enough, the approach builds new predictive models. The approach of (Camargo et al., 2022) lacks a drift detection mechanism for automatically detecting drops in performance. The authors do not specify how and when the predictions of the top-performing model are no longer considered as good enough and, hence, when the employed models need to be retrained. Furthermore, (Camargo et al., 2022) retrains its models on the entire available dataset, which implies that over time, these models are trained on a diminishing proportion of recent data and are consequently less and less sensitive to changes.

The study of (Miralles-Pechuán et al., 2023) is representative of machine learning approaches and com- pares batch and incremental algorithms using COVID-19 data from 50 countries. Results showed that incremental methods are more effective in adapting to changes and have lower computational cost compared to techniques such as LSTMs. A salient feature of (Miralles-Pechuán et al., 2023) is that it tests three different approaches. The first approach (*resp.* the second approach), involves training each model using data from a single country (*resp.* using data from the 50 countries). In the third approach, clustering is first applied to identify the most similar countries to the one being predicted and then each model is trained using data from those countries. Results showed that the third approach outperforms the two others. While the study of (Miralles-Pechuán et al., 2023) includes HT and HAT, surprisingly, it overlooks FIMT-DD. Similar to (Camargo et al., 2022), the work of (Miralles-Pechuán et al., 2023) does not include a drift detector for an automated management of the life cycle of machine learning models.

3. COLLABORATIVE DRIFT-DRIVEN REGRESSION (CDR)

The goal of a regression task is to learn a model M that predicts a real value, and not one of a discrete set of values as in classification. Formally, let S be a continuous stream of data: $S = \{(\overrightarrow{x^t}, y^t)\}$, where $\overrightarrow{x^t}$ is a feature vector, $y^t \in \mathbb{R}$ is the target variable and t the arrival timestamp. The goal is to incrementally learn $M: \vec{x} \rightarrow y$ as new data becomes available (Gomes et al., 2018). The predicted value of M is denoted as $\hat{y}$. When the actual value y gets revealed, the performance P is measured according to a loss function l: $P(M) = l(y, \hat{y})$. Performance of incremental models is typically measured using *prequential evaluation* (*a.k.a. test-then-train* evaluation), where each instance is used to test the model before it is used for training (Gomes et al., 2018). For the i^{th} instance: $\widehat{y^i} = M^{i-1}(\overrightarrow{x^i})$.

Our drift-driven collaborative regression approach CDR draws inspiration from the work of (Montiel et al., 2018a) on fast and slow classifiers. CDR combines incremental learning and batch learning to harness the strengths of both: (*i*) the accuracy of batch regressors, and (*ii*) the anytime property and adaptability to drifts of incremental regressors. Illustrated in Figure 1, the learning process of CDR involves the utilization of incremental learning to continuously and incrementally train and refine a regressor I, as new samples become available. Concurrently, batch learning is used to train a sequence of (batch) regressors $\{B_1, B_2, \ldots, B_n\}$. As illustrated in Figure 1, whenever a drift is detected, the current batch regressor B_i is invalidated and is subsequently replaced by a new regressor B_{i+1}. While I is trained on single samples as they arrive, a batch regressor B_i is trained on a (micro-) batch M_i containing the k most recent samples. This implies that the training process of B_{i+1} is deferred until k samples are gathered.

CDR uses *ADaptive WINdowing* (ADWIN) (Bifet & Gavaldà, 2007) for drift detection. The basic idea behind ADWIN is to maintain a variable-length sliding window W which increases in size as long as no drift is detected. To detect a drift, ADWIN repeatedly partitions W into two adjacent sub-windows W_0 and W_1 and compares their average to decide whether they are likely to originate from the same distribution. If W_0 and W_1 exhibit distinct enough averages and are of sufficient size, then a drift is detected and W is shrunk by dropping W_0 items from the window. In practice, ADWIN tests if the difference between the averages of W_0 and W_1 is larger than a variable value ϵ_{cut} computed as (Bifet & Gavaldà, 2007):

$$\epsilon_{cut} = \sqrt{\frac{1}{2m} ln \frac{4|W|}{\delta}}$$

, where m is the harmonic mean of W_0 and W_1. Unlike other existing drift detectors, ADWIN is assumption-free. Its only parameter is a confidence bound $\delta \in [0, 1]$, which enables adjusting the sensitivity to drifts.

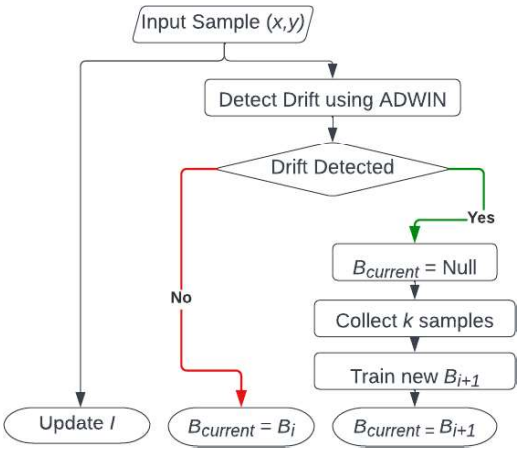


Figure 1. CDR - Learning Process

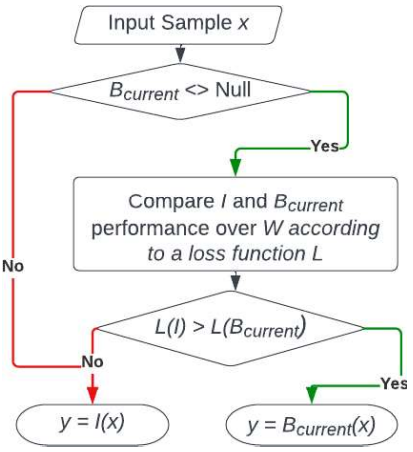


Figure 2. CDR - Inference Process

The inference process of CDR is depicted in the flowchart of Figure 2. As illustrated in Figure 2, when a batch regressor B_i is invalidated and until k samples are collected to train a new batch regressor B_{i+1}, only the incremental regressor I is used for inference. When a new batch regressor $B_{current}$ becomes available, CDR tracks the predictive performance of I and $B_{current}$ over a sliding window W containing the most recent observations. The top-performing model over W is then selected for inference. The aforementioned process is repeated for each new incoming instance. It is important to note that CDR is independent of the underlying learners and is compatible with any combination of incremental and batch methods.

4. EXPERIMENTAL EVALUATION

4.1 Tools and Datasets

We considered the dataset *Coronavirus Pandemic (COVID-19)* (Mathieu, E. et al., 2023), provided by Our World in Data (OWID). The original dataset includes daily information about the pandemic in 219 countries. Our target variables are the daily new confirmed cases and deaths per million people. We model the evolution of the daily new confirmed cases and deaths as function of the previously reported daily new confirmed cases per million people. For each country C and each record of C with a timestamp t, we consider the number of cases per million reported at 8 time points[1]: t minus 1 week, t minus 2 weeks,.., t minus 8 weeks. The obtained dataset contains nine input variables, ≈177k samples and covers the period starting from March 28, 2020 to November 30, 2022. When restricted to Tunisian data, the dataset contains ≈910 samples.

At the current state of our work, we implemented CDR using MOA (*Massive On line Analysis*) (Bifet et al., 2018), *Scikit-Multiflow* (Montiel et al., 2018b) and *Scikit-Learn* (Pedregosa et al., 2011). We configured CDR using FIMT-DD for incremental learning, DT (Pedregosa et al., 2011) for batch learning, and ADWIN for drift detection. All methods were run using their default settings. We compared the performance of batch and incremental models using a sliding window of one week and selected the best performing model to make predictions for the current input. Batch models were trained on windows of three weeks, with two weeks before and one week after a detected drift. The models' performance was evaluated using the Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE). In general, RMSE is more sensitive to large errors and outliers, while MAE is more interpretable.

4.2 Results and Discussion

A large number of experiments have been performed to demonstrate the effectiveness of CDR. Due to the lack of space, only few results are presented herein. In the first set of experiments, we compare the performance of CDR and established incremental methods using a prequential evaluation scheme. The obtained results are summarized in Table 1, and partially illustrated in Figures 3 and 4. As shown in Table 1, when compared to FIMT-DD, CDR achieves an improvement of 2.99%, 9.78%, 5.72% and 5.51% (resp. 44.59%, 63.97%, 15.75% and 16.41%) with regards to $RMSE_{Deaths}$, MAE_{Deaths}, $RMSE_{Cases}$ and MAE_{Cases} on world data (resp. single-country data). In contrast to HAT and FIMT-DD, HT lacks specialized mechanisms for handling drifts. When compared to HT, CDR achieves an improvement of 20.36%, 31.52%, 33.62% and 29.54% (resp. 57.32%, 79.41%, 26.12% and 30.7%) with regards to $RMSE_{Deaths}$, MAE_{Deaths}, $RMSE_{Cases}$ and MAE_{Cases} on world data (resp. single-country data). As we can observe in figures 3 and 4, CDR benefits from the continuous selection of predictions from both the incremental and the batch regressors. Near drift points, CDR opts in most cases for the batch model, which is only trained on recent data. However, the performance of the batch learner deteriorates quickly as we move away from drift points, primarily due to overfitting. Consequently, CDR transitions to the incremental learner. It's important to emphasize that the model selection near drift points is solely based on the performance (Mean Absolute Error) of both regressors, evaluated within a sliding window of one week.

[1] We do not consider the cases reported less than a week before t. Although including such observations could lead to more accurate models, they are not practical for timely policy responses.

ISBN: 978-989-8704-53-5 © 2023

Our second set of experiments compares the performance of a conventional batch regression tree against CDR. The prequential evaluation approach, commonly used in incremental learning, can be applied to batch learning by repeatedly retraining and reevaluating the model. For each training/testing round, the dataset is split into a training set and a test set in an order-preserving fashion. Instances used for testing the i^{th} batch model are appended to the training set of the $(i+1)^{th}$ model. The evaluation of the $(i+1)^{th}$ batch model is then performed on instances that arrived after its training and before the training of a new model. The process is repeated for multiple rounds until all the data has been used for both training and testing (except the first batch of data, which is only used for training, and the last batch, which is only used for testing). In (Miralles-Pechuán et al., 2023), training/evaluation rounds are referred to as *milestones*. In this study, we followed the aforementioned process (used also in (Miralles-Pechuán et al., 2023)) and adopted a realistic scenario where a new batch model is trained from scratch every ≈3 months, resulting in a set of 9 milestones (and, hence, 9 batch models). Tables 3 and 4 report the performance of the considered batch models and of CDR over the 9 milestones. As illustrated in Tables 3 and 4 CDR by far outperforms the corresponding batch model and respectively achieves an average improvement of 111.87%, 78.14%, 59.02% and 20.64% (resp. 124.83%, 90.29%, 109.17% and 65.13%) with regards to resp. $RMSE_{Deaths}$, MAE_{Deaths}, $RMSE_{Cases}$ and MAE_{Cases} on world data (resp. single-country data).

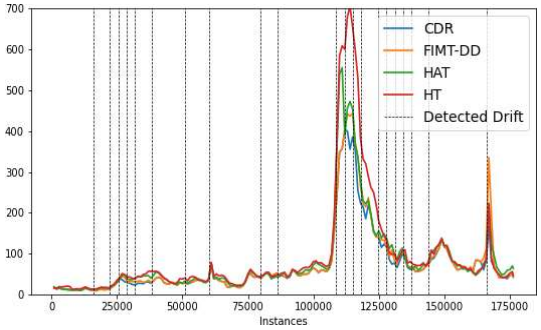


Figure 3. World - Daily New Confirmed Cases
MAE achieved by CDR and incremental learners

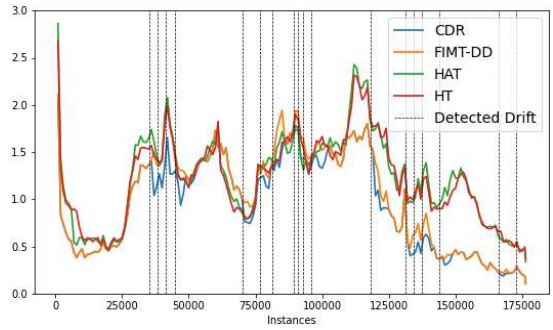


Figure 4. World - Daily New Confirmed Deaths
MAE achieved by CDR and incremental learners

Table 1. MAE and RMSE achieved by CDR and Incremental Learners

		Daily New Confirmed Deaths		Daily New Confirmed Cases	
		$RMSE_{Deaths}$	MAE_{Deaths}	$RMSE_{Cases}$	MAE_{Cases}
World	HT	2.01	1.21	174.1	91.51
	HAT	2.03	1.23	150.12	80.67
	FIMT-DD	1.72	1.01	137.75	74.53
	CDR	**1.67**	**0.92**	**130.29**	**70.64**
Tunisia	HT	2.47	2.44	61.56	58.71
	HAT	2.46	2.43	59.35	56.68
	FIMT-DD	2.27	2.23	56.50	52.29
	CDR	**1.57**	**1.36**	**48.81**	**44.92**

Besides confirming that CDR yields better results than those attained by each of the contributing models separately, the aforementioned experiments allow to draw the following important conclusions.

Retraining on recent data vs. retraining on the entire dataset. While (re-)training a batch model on the whole available dataset is beneficial in some cases to learn stable concepts or recurrent drifts, it is not the most suitable approach when it comes to predicting a pandemic's evolution. This can be observed in Tables 2 and 3, where in most cases, CDR becomes increasingly more efficient than the batch model over time. The reason behind this is that the batch model is (re-)trained on datasets with a diminishing proportion of recent data, causing it to become less and less sensitive to changes.

Learning using data from a single country vs. Learning using data from multiple countries. As it can observed in Tables 1 and 2 and 3, CDR exhibits higher improvements over incremental and batch models when training is performed using data from a single country. This is mainly due to the occurrence of drifts at different time points across countries, and to the fact that using data from multiple countries results in models that do not accurately represent any specific country. Overall, even though using data from multiple countries provides more training data, using data from a single country yields better results, as detecting (and, hence, adapting to) drifts is easier. It should be noted that this confirms the results found in (Miralles-Pechuán et al., 2023).

Table 2. Daily New Confirmed Deaths - MAE and RMSE achieved by CDR and the batch Decision Tree

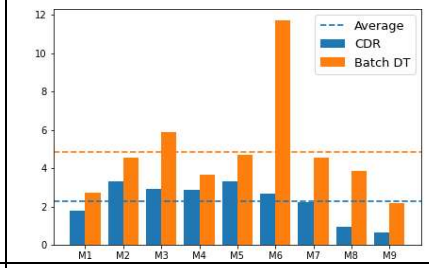
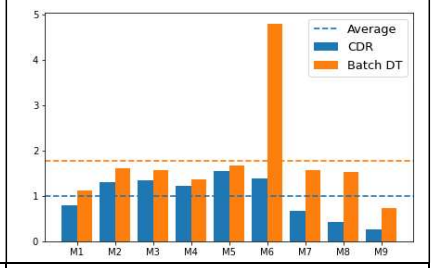
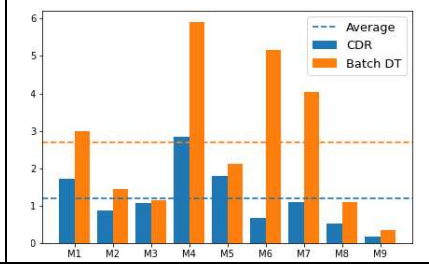
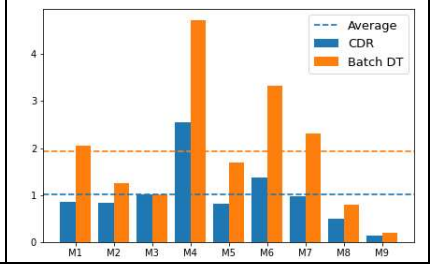
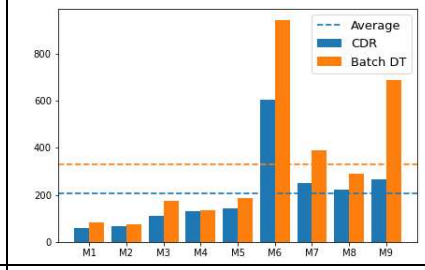
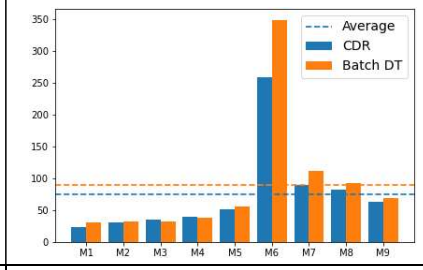
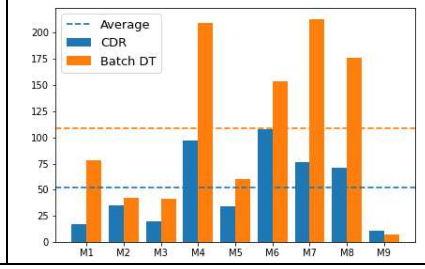
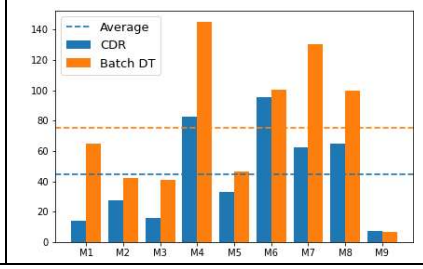
	Daily New Confirmed Deaths (per 1M)	
	$RMSE_{Deaths}$	MAE_{Deaths}
World		
Tunisia		

Table 3. Daily New Confirmed Cases - MAE and RMSE achieved by CDR and the batch Decision Tree

	Daily New Confirmed Cases (per 1M)	
	$RMSE_{Cases}$	MAE_{Cases}
World		
Tunisia		

5. CONCLUSION

Pandemics will never cease to emerge and threaten both public health and the global economy. Therefore, it is crucial to learn from past pandemics to develop effective tools for preventing and controlling future outbreaks. This involves not only researching treatments and therapies, but also implementing efficient epidemiological surveillance systems. In this paper we proposed CDR, an innovative collaborative regression strategy for predicting the evolution of pandemics. CDR continuously refines the underlying incremental model as data arrives, uses ADWIN as drift detector and retrains a new batch model on recent observations whenever it detects a drift. At the current state of our work, we implemented CDR using FIMT-DD as incremental regressor and the Decision Tree as batch regressor. Experiments on COVID-19 data, showed that CDR is an effective collaboration strategy that yields better results than those attained by the incremental and the batch model separately.

In this paper, we mainly focused on connecting the dots between batch regressors and incremental regressors and on the relevance of using them in conjunction to predict the evolution of a pandemic. The results obtained are highly encouraging for exploring other forms of collaboration between batch learning and incremental learning. a part of our future work, we intend to investigate the use of incremental and batch ensemble models instead of single models to further alleviate the effects of concept drifts. On the other hand, our experimental study revealed that even though using data from multiple countries provides more training data, using data from a single country yields better results. This is mainly due to the occurrence of drifts at different time points across countries. In our future work, we intend to identify subgroups of countries exhibiting similar patterns and apply CDR to each subgroup independently. We believe that doing so will allow to leverage more training data while maintaining the ability to efficiently detect and adapt to drifts.

REFERENCES

Bifet, A. et al., 2018. *Machine learning for data streams: with practical examples in MOA*. MIT Press, Massachusetts, USA.

Bifet, A. and Gavaldà, R., 2007. Learning from time-changing data with adaptive windowing. *Proceedings of the 2007 SIAM International Conference on Data Mining (SDM)*. pp. 443–448.

Camargo, E. et al., 2022. An incremental learning approach to prediction models of seird variables in the context of the covid-19 pandemic. *In Health and Technology*, Vol. 12, No. 4, pp. 2190–7196.

Domingos, P. and Hulten, G., 2000. Mining high-speed data streams. *Proceedings of the sixth ACM SIGKDD international conference on Knowledge Discovery and Data mining*. Boston, USA, pp. 71–80.

Gomes, H. M. et al., 2018. Adaptive random forests for data stream regression. *26th European Symposium on Artificial Neural Networks (ESANN 2018)*, Bruges, Belgium, pp. 267–272

Ikonomovska, E. et al., 2011. Learning model trees from evolving data streams. *Data mining and knowledge discovery*, Vol. 23, No. 1, pp. 128–168.

Marani, M. et al., 2021. Intensity and frequency of extreme novel epidemics. *Proceedings of the National Academy of Sciences*, Vol. 118, No. 35, e2105482118.

Mathieu, E. et al., 2023. OurWorldInData.org. *Coronavirus Pandemic (COVID-19)*. Retrieved from: "https://ourworldindata.org/coronavirus" [Online Resource].

Miralles-Pechuán, L. et al., 2023. Forecasting covid-19 cases using dynamic time warping and incremental machine learning methods. *Expert Systems*, Vol. 40, No. 6, e13237.

Montiel, J. et al., 2018a. Learning fast and slow: A unified batch/stream framework. *2018 IEEE International Conference on Big Data*, Seattle, WA, USA, pp. 1065–1072.

Montiel, J. et al., 2018b. Scikit-multiflow: A multi-output streaming framework. *Journal of Machine Learning Research*, Vol. 19, No. 72, pp. 1–5.

Mouss, H. et al., 2004. Test of page-hinckley, an approach for fault detection in an agro-alimentary production system. *2004 5th Asian Control Conference (IEEE Cat. No. 04EX904)*, Melbourne, VIC, Australia, pp. 815–818.

Pedregosa, F. et al., 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, Vol. 12, pp. 2825–2830.

Salah, I. et al., 2023. On the use of data augmentation for stance and fake news detection. *Journal of Information and Telecommunication*, Vol. 7, No. 3, pp. 359–375.

.

C.2 Integrating Deep and Handcrafted Features for Enhanced Remote Sensing Image Classification

Vian Abdulmajeed Ahmad, Khaled Jouini, Amel Tuama & Ouajdi Korbaa
21^{st} ACS/IEEE International Conference on Computer Systems and Applications (AICCSA).
Octobre 2024.

https://aiccsa.net/AICCSA2024/wp-content/uploads/2024/10/AICCSA2024_program_formatKBv1910vf.pdf
(Track 6, Session 1, Multimedia, Computer Vision, and Image Processing)
CORE Rank : C

ICORE Conference Portal

Back to search

ACS/IEEE International Conference on Computer Systems and Applications

Acronym: AICCSA

DBLP Source: https://dblp.uni-trier.de/db/conf/aiccsa

Source: CORE2023

Rank: C

Field Of Research: 46 - Information and Computing Sciences (h-index) (citation)

Source: CORE2021

Rank: C

Field Of Research: 46 - Information and Computing Sciences

Requested as B, retained as C (Data 1) (Decision)

Source: CORE2020

Rank: C

Field Of Research: 46 - Information and Computing Sciences

Source: CORE2018

Rank: C

Field Of Research: 08 - Information and Computing Sciences†

Field Of Research: 12 - Built Environment and Design†

Source: CORE2017

Rank: C

Field Of Research: 08 - Information and Computing Sciences†

Field Of Research: 12 - Built Environment and Design†

The ACS/IEEE 21st International Conference on Computer Systems and Applications (AICCSA 2024)

22-26 October 2024 – Sousse, Tunisia

Co-Sponsored by the Arab Computer Society (ACS) and IEEE Computer Society

Certificate of Contribution

This is to certify that the following paper was presented during the 21st International Conference on Computer Systems and Applications (AICCSA 2024) held from October 22-26, 2024.

173

Paper Title: Integrating Deep and Handcrafted Features for Enhanced Remote Sensing Image Classification

Author: **Khaled Jouini**

General Chairs

Takoua Abdellatif, University of Sousse, Tunisia

Cihan Tunc, University of North Texas, USA

Sami Yangui, LAAS-CNRS, Toulouse, France

المدرسة التونسية للتقنيات
Ecole Polytechnique de Tunisie

HORIZON SCHOOL OF DIGITAL TECHNOLOGIES

ESGITECH ECOLE SUPERIEURE DE GENIE INFORMATIQUE ET DE TECHNOLOGIE
L'université des métiers de demain

Integrating Deep and Handcrafted Features for Enhanced Remote Sensing Image Classification

Vian Abdulmajeed Ahmed
MARS Research Lab, LR17ES05
ISITCom, University of Sousse
Sousse, Tunisia
orcid.org/0009-0002-5924-6139

Khaled Jouini
MARS Research Lab, LR17ES05
ISITCom, University of Sousse
Sousse, Tunisia
orcid.org/0000-0001-5049-4238

Amel Tuama
Computer Engineering Techniques Department
Northern Technical University
Mosul, Iraq
orcid.org/0000-0002-3802-9074

Ouajdi Korbaa
MARS Research Lab, LR17ES05
ISITCom, University of Sousse
Sousse, Tunisia
orcid.org/0000-0003-4462-1805

***Abstract*—Satellite imagery supports critical applications such as land cover mapping, environmental monitoring, disaster assessment, and urban planning. Despite significant advancements, challenges in analyzing satellite imagery persist, primarily due to data variability, atmospheric conditions, and complex land cover patterns. Traditional handcrafted descriptors like Scale-Invariant Feature Transform (SIFT) and encoding techniques such as Bag-of-Visual-Words (BoVW) are effective but often fall short in capturing global context and spatial relationships due to their inherent local nature. The advent of deep learning (DL), propelled by ample data and computational resources, has markedly improved satellite image analysis. However, the reliance on extensive annotated data constrains the wider applicability of DL methods.**

This study harnesses the strengths of both deep and handcrafted features to enhance the classification accuracy of remote sensing images. Specifically, we synergize SIFT descriptors with pretrained MobileNetV2 and VGG16 deep features. While SIFT excels in capturing local features essential for identifying specific image characteristics, pretrained DL models provide enriched representations with global context, spatial relationships, and hierarchical features. This integration aims to overcome the individual limitations of each method, enabling the model to effectively handle perturbations, scale variations, and diverse landscapes. Extensive evaluations on the EuroSAT dataset demonstrate that our approach outperforms, not only SIFT, VGG16, and MobileNetV2 when used separately, but also surpasses state-of-the-art remote sensing image classification approaches. Another salient advantage of our approach is its robust applicability in scenarios with limited labeled data — a prevalent challenge in remote sensing image classification.

***Index Terms*—Remote Sensing, Land Cover Mapping, Features Fusion, Transfer Learning, Scale-Invariant Feature Transform (SIFT), Image Classification.**

I. INTRODUCTION

Satellite imagery offers a unique and comprehensive perspective of the Earth's surface, playing a crucial role in various applications such as land cover mapping, environmental monitoring, disaster response and urban planning [2]. The increasing volume of remote sensing images, fuelled by advancements in Earth observation, presents a critical yet challenging task: extracting valuable information from these extensive and intricate datasets [13]. *Scene and image classification*, which involves assigning predefined semantic classes to remote sensing images, lies at the heart of this challenge. Effective remote sensing image classification requires the capability to discern complex spatial patterns, while maintaining robustness against variations in scale, atmospheric conditions, and noise [23].

Early methods in remote sensing image classification heavily relied on manually designed descriptors, represented by the widely used *Scale-Invariant Feature Transform* (SIFT) [14]. SIFT and similar approaches excel in capturing distinctive local features, such as corners, edges, and textured regions. However, due to their inherent local nature and inability to directly represent the entirety of scenes, SIFT and similar approaches often struggle to adequately capture the complex spatial and contextual information present in remote sensing images [5].

The advent of deep learning, coupled with the increased availability of data, has brought about a paradigm shift in remote image classification. Deep learning models, trained on vast datasets containing millions of labeled images, enable feature extraction capabilities and levels of accuracy beyond the reach of traditional methods [5]. Nevertheless, the data-hungry nature of deep learning limits its scope of application. To mitigate the requirement for vast amounts of annotated training data, *transfer learning* has emerged as a strategic solution. Transfer learning involves harnessing knowledge gained from one task and applying it to a related but different task [20]. In the context of remote sensing image classification, transfer learning entails using a deep learning model pretrained on a large and general-purpose dataset like ImageNet, and fine-tuning it on a smaller task-specific remote sensing dataset. VGG16 [18] and MobileNetV2 [17] are two notable pretrained models frequently employed in this context.

This study leverages the complementary strengths of handcrafted and deep features to enhance the model's performance

in diverse scenes. We specifically utilize SIFT for its proven effectiveness in capturing unique local details crucial for identifying specific image characteristics. In contrast, the deep features extracted from VGG16 and MobileNetV2 contribute global context, spatial relationships, and hierarchical features, complementing the model's understanding of the entire scene. By fusing these distinct yet complementary feature sets, we aim to improve the model's resilience, robustness, and generalizability across diverse landscapes and scenarios. Such enhanced generalizability is crucial, as real-world scenes encompass a broad spectrum, ranging from urban to rural landscapes. Experiments conducted on the real-world EuroSAT dataset [10] demonstrate that our approach outperforms, not only SIFT, VGG16, and MobileNetV2 when used separately, but also surpasses state-of-the-art remote sensing image classification approaches.

The remainder of this paper is organized as follows. Section II provides a concise overview of previous research, Section III introduces our features fusion approach. Section IV presents a comparative experimental analysis using the EuroSAT dataset. Finally, Section V concludes the paper.

II. RELATED WORK

The EuroSAT dataset [10], used in our experimental study, is a widely recognized and extensively used dataset for *Land Use and Land Cover* (LULC) classification. This dataset, sourced from the Copernicus Earth observation program, consists of 27,000 geotagged image patches, each measuring 64x64 meters and having a spatial resolution of 10 meters. The dataset comprises ten distinct classes, with each class consisting of 2,000 to 3,000 images. As depicted in Figure 1, the classes encompass a range of land use and land cover types, including permanent and annual crops, pastures, rivers, seas and lakes, forests, herbaceous vegetation, industrial and residential buildings, and highways. For the sake of conciseness and due to lack of space, we mainly focus in the sequel on approaches presenting similarities with our work or that use the EuroSAT dataset. Existing approaches and studies can be broadly classified into two families: Machine Learning (ML)-based algorithms [4], [12], [22] and Deep Learning (DL)-based methods [9], [15], [16].

Studies by Hu et al. [12], Chen & Tian [4], and Thakur & Panse [22] are representative of ML-based approaches. [12] proposed a method utilizing randomly sampled image patches for Unsupervised Feature Learning (UFL) and applied *Bag-of-Visual-Words* (BoVW) [6] encoding to UFL. Experiments conducted on an aerial scene dataset present encouraging results with an accuracy of 90.03%. [4] introduced the Pyramid of Spatial Relations (PSR) model, designed to incorporate both relative and complete spatial information into the BOVW framework. Experiments conducted on a high-resolution remote sensing image revealed that the PSR model achieves an average classification accuracy of 89.1%. In [22] four ML algorithms were evaluated for their performance in the LULC classification task: Decision Tree (DT), K-Nearest Neighbour (KNN), Support Vector Machine (SVM), and Random Forest (RF). The results from [22] highlighted the superior performance of RF compared to DT, KNN, and SVM, with SVM and DT exhibiting similar levels of effectiveness.

Fig. 1. Sample Images extracted from the EuroSAT Dataset [11]

The studies [10], [8], and [21] represent DL-based approaches. In [21], the authors introduce an interpretable DL framework for LULC classification using SHapley Additive exPlanations (SHAPs). They employ a compact CNN model for image classification, followed by feeding the results to a SHAP deep explainer, achieving an accuracy of 94.72% on

the EuroSAT dataset. The approach in [8] utilizes different CNN architectures for feature extraction, including VGG19, ResNet50, and InceptionV3. These extracted features are then recalibrated using the Channel Squeeze & Spatial Excitation (sSE) block, with Twin SVM (TWSVM) serving as classifiers, achieving a 94.39% F1-score on the EuroSAT dataset. In [10], various CNN architectures were compared, including a shallow CNN, a ResNet50-based model, and a GoogleNet-based model. The achieved classification accuracies on the EuroSAT dataset were 89.03%, 98.57%, and 98.18%, respectively. Additionally, [10] evaluated the performance of SVM using SIFT features and BOVW encoding, revealing that all CNN models outperformed the SIFT-SVM method.

As outlined in this section, existing approaches typically concentrate on either classical ML or DL methods, without exploring the potential benefits of their integration. In a prior work [2], we demonstrated that fusing SIFT descriptors with features extracted from a straightforward Convolutional Neural Network led to improved performance compared to using each method separately. However, the achieved accuracy of 92% did not match the levels reached by top-performing pretrained DL models. In this study, we extend this investigation by employing transfer learning and examining the integration of more advanced deep features with conventional SIFT descriptors.

III. SYNERGIZING HANDCRAFTED AND DEEP FEATURES

In this section, we introduce a novel approach for remote sensing image classification that synergizes handcrafted SIFT descriptors [19] with advanced deep features. We chose MobileNetV2 [17] for its efficiency and ability to operate within bandwidth constraints, making it particularly suitable for real-time satellite image processing. VGG16 [3] was selected due to its deep architectural layers, which are highly effective in extracting intricate patterns from high-resolution images—a crucial factor for distinguishing subtle differences in land cover. This hybrid approach not only leverages the robust local feature detection capabilities of SIFT but also complements them with the global contextual understanding provided by deep models. Such integration is designed to address specific gaps in traditional classification methods, especially those encountered in handling complex and variable imagery typical of remote sensing applications.

The framework of the proposed approach is depicted in Figure 2. To evaluate and quantify the benefits gained by combining handcrafted and deep features, we consider five distinct models: a SIFT-based model, pretrained MobileNetV2 fine-tuned on the EuroSAT dataset, pretrained VGG16 fine-tuned on the EuroSAT dataset, a hybrid model combining SIFT descriptors with MobileNetV2 deep features, and another hybrid model combining SIFT descriptors with VGG16 deep features. The details of each model are presented in the following subsections.

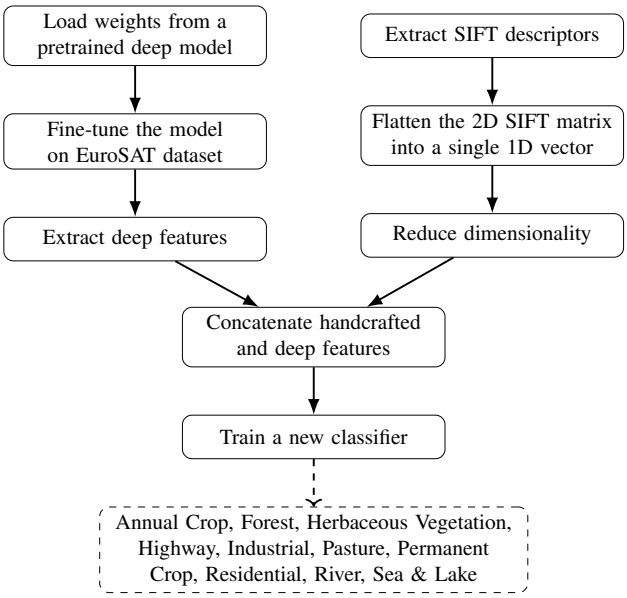


Fig. 2. Overview of the proposed approach

A. Handcrafted- Based Classfication

The SIFT-based classification used in our work follows the same steps as in [2], [10]. As illustrated in Figure 3, the process begins with converting the original satellite images to grayscale format. This conversion simplifies the data while retaining essential visual characteristics. Grayscale images contain intensity values ranging from black to white, representing different levels of brightness, which are crucial for identifying key features in the images. Once the images are in grayscale, the SIFT algorithm is applied to identify *keypoints* and extract *local feature descriptors*. The keypoints identified by the SIFT algorithm are areas in the image with significant variations in intensity across different directions. These keypoints are typically found at corners, edges, or distinct texture patterns (features that are likely to be stable across varying conditions) [14]. In the context of satellite images, keypoints often correspond to transitions between different land covers, such as the edges of roads, boundaries of fields, or distinct changes in topography.

Once keypoints are identified, SIFT computes a descriptor for each of them. This descriptor encapsulates information about the gradients or directional changes in intensity surrounding that keypoint within a localized patch of the image. The standard SIFT descriptor is generated by creating a histogram of gradient orientations within this patch, divided into a 4×4 grid, with each of the 16 cells contributing eight orientation bins, resulting in a 128-element descriptor vector. This descriptor effectively captures the local texture and shape around the keypoint, making it distinctive and easy to compare across images [14].

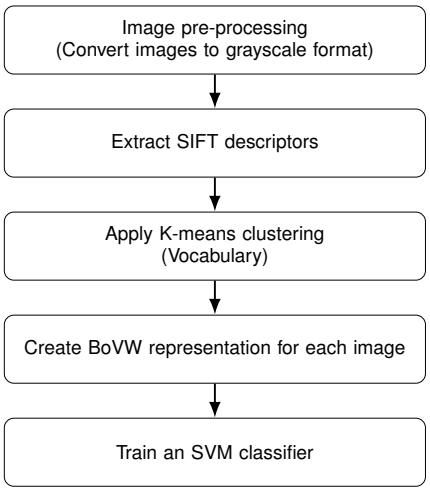


Fig. 3. SIFT-based model construction

A key feature of SIFT descriptors is their invariance to changes in scale and rotation. During detection, the algorithm identifies potential keypoints at multiple scales using a set of Gaussian blurred images, which are generated in an image pyramid. Each keypoint's scale corresponds to the level of the pyramid where it was detected, ensuring that the same feature can be recognized at different sizes [14]. Additionally, orientation assignment, based on the local image gradient, further ensures that the descriptor is rotation invariant, allowing the keypoints to be robustly matched across different images despite variations in viewpoint, zoom level, or illumination [14].

To encode the SIFT descriptors for efficient classification, we adopt the Bag-of-Visual-Words (BoVW) technique. In this stage, the extracted descriptors are *quantized* into *visual words*. This quantization involves using the k-means clustering algorithm to group similar SIFT descriptors into clusters based on their feature vectors. The centroids of these clusters become *visual words*. Subsequently, images are encoded in accordance with these visual words, yielding a histogram-like representation that succinctly captures the visual content ($i.e.$ each cluster is represented by a visual word, and each image is described by the count of each visual word it contains). The final step in the SIFT-based model construction process involves training a Support Vector Machine (SVM) model on the BoVW-represented satellite images.

B. Transfer Learning and Deep Features-Based Classfication

The *convolutional layers* of deep models like MobileNetV2 and VGG16 operate by applying learned filters that slide across the image. Each filter ($i.e.$ a small grid of weights), interacts with corresponding image patches. As the filter slides across the image, these weights are element-wise multiplied with pixel values, and the results are summed to create a *feature map*. Convolutional filters possess two key properties: size and learned weights. The size of the filter determines the scale of detectable patterns, with larger filters targeting bigger objects. The weights, learned during training, enable the filters to become sensitive to specific image features ($e.g.$ edges, corners, color combinations, etc.). A convolutional layer typically uses a set of diverse filters to detect a variety of patterns simultaneously. The inherent design of filters enables them to be sensitive to patterns regardless of their orientation (within a limited range).

As convolutional layers are stacked, deep models learn increasingly complex patterns: early layers typically capture basic elements such as edges and corners and subsequent layers interpret these elements to recognize objects like buildings or fields and understand their spatial relationships. This *hierarchical learning* process enables deep models to capture not only basic shapes and textures but also how these elements combine to form objects and entire scenes. This capability is crucial in remote sensing, where the accurate classification of land cover types depends on both high-detail visual cues and the broader context of the scene.

The architecture of MobileNetV2, partially illustrated in Figure 4, leverages depthwise separable convolutions within bottleneck blocks to achieve efficient feature extraction. Depthwise convolutions capture per-channel features, followed by pointwise convolutions for dimensionality reduction. Bottleneck blocks further enhance efficiency through the use of 1×1 pointwise convolutions for channel reduction and expansion, along with residual connections. As the network progresses, both filter sizes and channel numbers increase, allowing for the extraction of increasingly complex features. *Global Average Pooling* (or GAP), replacing traditional flattening, captures global features while preserving spatial information. Following the feature extraction stages, MobileNetV2 employs activation functions throughout the network to introduce non-linearity. The ReLU (Rectified Linear Unit) activation function, which allows only positive values to pass through, is used in MobileNetV2. Finally, the extracted features are fed into fully-connected (dense) layers that perform linear transformations on the features, ultimately leading to a final output layer with the desired number of classes (10 for EuroSAT classification). The SoftMax activation function in the final layer generates class probabilities, providing a measure of confidence for each predicted class. Throughout the training phase, optimizers and loss functions (like Adam and sparse categorical cross-entropy) are employed to optimize the model's performance.

The architecture of VGG16 follows a similar pattern of convolutional and pooling layers as MobileNetV2 but with a significantly deeper structure. It relies on repeated stacks of small 3×3 convolutional filters followed by max pooling for feature extraction. The number of filters progressively doubles from 64 to 512 as the network deepens, allowing VGG16 to capture increasingly complex features at increasing depths. Subsequent max-pooling layers contribute to the extraction of hierarchical features by downsampling the feature maps, emphasizing the most salient features. While not as computationally efficient as some modern architectures, VGG16's

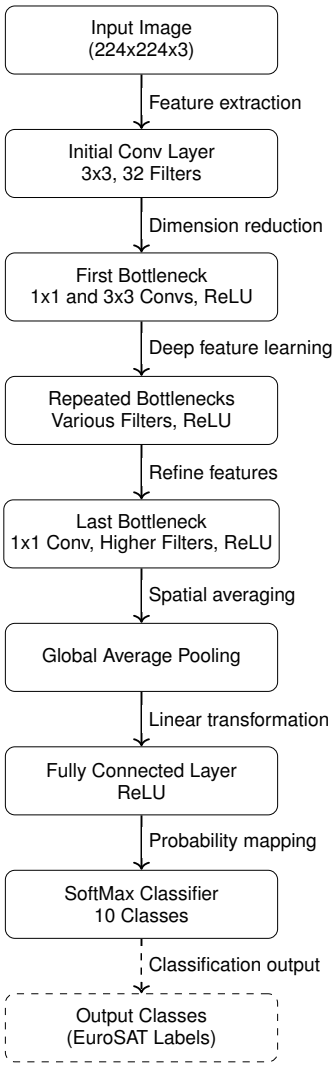


Fig. 4. Architecture of the fine-tuned MobileNetV2 model

depth and systematic design make it well-suited for intensive image recognition tasks [8].

Instead of training from scratch, in this study, we leverage transfer learning by utilizing pretrained VGG16 and MobileNetV2 models with weights acquired from the general-purpose ImageNet dataset. These models are then fine-tuned on the EuroSAT dataset. During fine-tuning, the initial layers of the pretrained models are frozen, while the later layers are trained on the EuroSAT data to learn features specific to the LULC classification task.

C. Handcrafted and Deep Features: A Hybrid Approach

This paper proposes a hybrid approach that combines handcrafted SIFT descriptors with features learned from deep models. By leveraging transfer learning and combining these complementary feature sets, we aim to capitalize on the

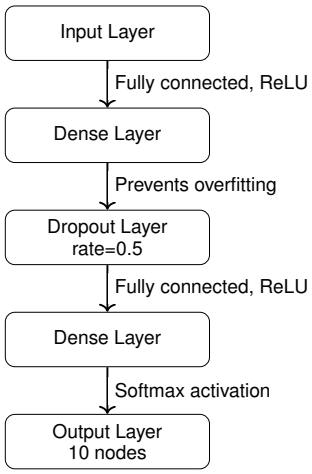


Fig. 5. Neural Network Architecture for Hybrid SIFT-MobileNetv2 Features

strengths of both approaches. SIFT can potentially capture those crucial local details that deep models might overlook, such as specific textures. Learned deep features, on the other hand, provide a broader understanding of the scene's semantic content, encompassing the relationships between different objects and regions. This complementary nature of SIFT and deep learning features makes feature fusion a promising strategy for creating a more comprehensive and informative representation for remote sensing image classification tasks. The combination process includes the following substeps:

1) *Preprocessing*: Initially, the input images, which possess varying dimensions, undergo preprocessing and resizing to adhere to the (224, 224, 3) input shape specifications required by the fine-tuned MobileNetV2 and VGG16 models.
2) *Feature Extraction*: SIFT descriptors are extracted for each image in the dataset. Simultaneously, deep features are extracted from the output of the last convolutional layer before the fully connected layers of the fine-tuned VGG16 model (this is achieved by setting the parameter "top" to False). In the case of the fine-tuned MobileNetV2, deep features are extracted from the output of the Global Average Pooling layer.
3) *Feature Concatenation*: SIFT identifies potentially hundreds or thousands of keypoints per image. For each keypoint, SIFT calculates a descriptor, which is a vector containing information about the surrounding image patch centered on the keypoint. The resulting high-dimensional matrix produced by SIFT is not directly suitable for machine learning algorithms, which typically require a single fixed-length feature vector per image. To simplify data representation and ensure compatibility with most machine learning algorithms, all individual keypoint descriptors are concatenated into a single row vector (flattening). Subsequently, these flattened SIFT descriptors are horizontally concatenated

TABLE I
PRECISION, RECALL, F1-SCORE ACHIEVED BY THE SIFT-BASED MODEL

Class	Precision	Recall	F1-Score
AnnualCrop	0.65	0.71	0.68
Forest	0.00	0.00	0.00
HerbaceousVegetation	0.40	0.36	0.38
Highway	0.53	0.45	0.49
Industrial	0.74	0.86	0.79
Pasture	0.28	0.65	0.40
PermanentCrop	0.51	0.41	0.45
Residential	0.77	0.78	0.78
River	0.42	0.29	0.35
SeaLake	0.00	0.00	0.00

TABLE II
PRECISION, RECALL, F1-SCORE ACHIEVED BY THE FINE-TUNED PRETRAINED VGG16 MODEL

Class	Precision	Recall	F1-Score
AnnualCrop	0.84	0.88	0.86
Forest	0.96	0.95	0.95
HerbaceousVegetation	0.76	0.80	0.78
Highway	0.97	0.88	0.88
Industrial	0.63	0.70	0.66
Pasture	0.81	0.83	0.82
PermanentCrop	0.69	0.73	0.71
Residential	0.89	0.75	0.72
River	0.90	0.65	0.72
SeaLake	0.98	0.97	0.97

with the extracted deep features. This process generates a unified feature vector containing information from both sources.

4) *New Classifier Training*: The fused feature vectors, along with their corresponding class labels from the EuroSAT dataset, are used to train a new classifier. For the hybrid SIFT-VGG16 model, we employed a simple Support Vector Machine (SVM) with a radial basis function (RBF) kernel. In the case of the hybrid SIFT-MobileNetV2, we employed a straightforward neural network classifier (Figure 5) [1].

TABLE III
PRECISION, RECALL, F1-SCORE ACHIEVED BY THE FINE-TUNED PRETRAINED MOBILENETV2 MODEL

Class	Precision	Recall	F1-Score
AnnualCrop	0.97	0.98	0.97
Forest	0.97	0.98	0.98
HerbaceousVegetation	0.96	0.96	0.96
Highway	0.98	0.94	0.94
Industrial	0.95	0.98	0.98
Pasture	0.95	0.94	0.95
PermanentCrop	0.95	0.95	0.95
Residential	0.98	0.98	0.98
River	0.96	0.96	0.96
SeaLake	0.99	0.99	0.99

TABLE IV
PRECISION, RECALL, F1-SCORE ACHIEVED BY THE HYBRID SIFT-VGG16 MODEL

Class	Precision	Recall	F1-Score
AnnualCrop	0.91	0.97	0.94
Forest	0.88	0.93	0.90
HerbaceousVegetation	0.89	0.90	0.89
Highway	0.90	0.87	0.89
Industrial	0.93	0.96	0.94
Pasture	0.90	0.89	0.89
PermanentCrop	0.90	0.87	0.88
Residential	0.97	0.97	0.97
River	0.91	0.88	0.89
SeaLake	0.96	0.88	0.92

TABLE V
PRECISION, RECALL, F1-SCORE ACHIEVED BY THE HYBRID SIFT-MOBILENETV2 MODEL

Class	Precision	Recall	F1-Score
AnnualCrop	0.99	0.99	0.99
Forest	0.99	0.98	0.98
HerbaceousVegetation	0.98	0.97	0.98
Highway	0.99	0.99	0.99
Industrial	0.98	0.98	0.98
Pasture	0.97	0.98	0.98
PermanentCrop	0.98	0.98	0.98
Residential	0.99	0.99	0.99
River	0.98	0.98	0.98
SeaLake	0.99	0.98	0.99

IV. EXPERIMENTAL STUDY

Our experimental study primarily aims to quantify the accuracy gains achieved through the fusion of handcrafted and deep features. We therefore compared the remote sensing image classification models detailed in section III. The SIFT-based model was implemented and evaluated using OpenCV [7]. The pretrained MobileNetV2 and VGG-16 models were fine-tuned on the EuroSAT dataset and evaluated using Tensorflow [1]. All reported evaluation metrics were obtained from the same test set. The experiments were conducted with default settings to explore potential benefits of feature fusion without introducing additional bias.

Tables I to V present the per-class recall, precision, and F1-score achieved by each studied model. As expected and shown in these tables and in Table VI, the SIFT-based model is by far outperformed by the fine-tuned pretrained models: VGG16 and MobileNetV2 models respectively exhibit improvements of 48.21% and 64.29% in accuracy with regards to the SIFT-based model. This underscores the benefits derived from deep learning and transfer learning. Notably, the fine-tuned pretrained MobileNetV2 model achieves an impressive accuracy of 97%.

As shown in Table VI, our findings also reveal substantial accuracy enhancements achieved by the fusion models compared to individual methods. Specifically, the Hybrid SIFT-VGG16 model demonstrates accuracy improvements of

[1]We evaluated the performance of the fused feature vectors with various basic machine learning algorithms and retain the best performing models in each case.

TABLE VI
ACCURACY ACHIEVED BY THE STUDIED MODELS

Model	Accuracy
SIFT-based model	0.56
Fine-tuned pretrained VGG16	0.83
Fine-tuned pretrained MobileNetV2	0.97
Hybrid SIFT with fine-tuned pretrained VGG16	0.93
Hybrid SIFT with fine-tuned pretrained MobileNetV2 SIFT	0.98

TABLE VII
ACCURACY ACHIEVED BY OUR HYBRID APPROACH COMPARED TO EXISTING REMOTE SENSING IMAGE CLASSIFICATION APPROACHES (ON EUROSAT DATASET)

Model	Accuracy
SIFT-BoVW (k = 500) [10]	0.7
UFL [12]	0.9
CNN two layers [10]	0.87
Pyramid of spatial relations [4]	0.89
SIFT-CNN two Layers [2]	0.92
Combination of different CNNs deep architectures [8]	0.94
Fine-tuned pretrained ResNet50 [11]	0.98
Fine-tuned pretrained GoogleNet [11]	0.98
Hybrid SIFT with fine-tuned pretrained VGG16	0.93
Hybrid SIFT with fine-tuned pretrained MobileNetV2	0.98

66.07% over the SIFT-based model and 12.05% over the fine-tuned pretrained VGG16 model. Similarly, the Hybrid SIFT-MobileNetV2 model exhibits accuracy improvements of 76.79% and 20.62% over the SIFT-based and the fine-tuned pretrained MobileNetV2 models, respectively. These results underscore the potential of combining handcrafted and deep learning features for remote sensing image classification. By leveraging the complementary strengths of both approaches, our fusion models achieve significantly higher accuracy levels. This may be attributed to SIFT capturing local details that deep models might have missed, while deep features provide broader scene context, leading to improved performance. The exceptional accuracy of the hybrid SIFT-MobileNetV2 fusion model, reaching 99%, highlights its effectiveness in learning input data features and generalizing well to unseen samples.

Table VII provides a comparison of our models' accuracy with state-of-the-art remote sensing classification approaches. As shown in Table VII, our hybrid models achieve a significant accuracy improvement compared to ML-based models and exhibit competitive performance compared to DL-based approaches.

V. CONCLUSION AND FUTURE WORK

This paper proposes a novel approach for enhancing remote sensing images classification using two complementary feature extraction methods: SIFT and pretrained deep models. While deep features extracted from VGG16 and MobileNetV2 excel in capturing global contexts and spatial relationships, handcrafted features extracted from SIFT excel in capturing unique local details that deep models might overlook. By synergizing these distinct and complementary feature sets, we aim to enhance the model's resilience, robustness, and generalization capabilities across diverse landscapes and scenarios. Such enhanced generalization is crucial, given that real-world scenes encompass a broad spectrum, ranging from urban to rural landscapes. Experiments conducted on the real-world EuroSAT dataset demonstrate that our approach outperforms not only individual implementations of SIFT, VGG16, and MobileNetV2 but also surpasses existing remote sensing image classification approaches.

In this study, we have investigated the integration of conventional and deep learning approaches at an early stage, specifically focusing on feature extraction. However, there are additional forms of fusion, namely late and hybrid fusion, that we intend to explore as part of our future work. Late fusion involves merging the outputs of independently trained classifiers at a later stage, potentially at the decision level. This strategy allows for the utilization of diverse feature representations and model architectures, which could enhance classification performance. Conversely, hybrid fusion strategies aim to seamlessly integrate both conventional and deep features within a unified framework, leveraging the complementary strengths of each approach. Through further exploration of late and hybrid fusion strategies, we aim to uncover novel insights and potentially enhance the effectiveness of remote sensing image classification methods.

REFERENCES

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citron, G. Corrado, A. Davis, J. Dean, M. Devin, and et. al. Tensorflow: Large-scale machine learning on heterogeneous systems, 2015.

[2] Vian Abdulmajeed Ahmad, Khaled Jouini, and Ouajdi Korbaa. A fusion approach for enhanced remote sensing image classification. In *19th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, VISIGRAPP*, Italy, 2024.

[3] R. G. Babu, K. U. Maheswari, C. Zarro, B. D. Parameshachari, and S. L. Ullo. Land-use and land-cover classification using a human group-based particle swarm optimization algorithm with an lstm classifier on hybrid pre-processing remote-sensing images. *Remote Sens.*, 12(24):1–28, 2020.

[4] S. Chen and Y. Tian. Pyramid of spatial relatons for scene-level land use classification. *IEEE Trans. Geosci. Remote Sens.*, 53(4):1947–1957, 2015.

[5] G. Cheng, X. Xie, J. Han, L. Guo, and G. S. Xia. Remote sensing image scene classification meets deep learning: Challenges, methods, benchmarks, and opportunities. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, 13(X):3735–3756, 2019.

[6] Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004.

[7] I. Culjak, D. Abram, T. Pribanic, H. Dzapo, and M. Cifrek. A brief introduction to opencv. In *2012 Proceedings of the 35th International Convention MIPRO*, page 1725–1730, 2012.

[8] H. I. Dewangkoro and A. M. Arymurthy. Land use and land cover classification using cnn, svm, and channel squeeze & spatial excitation block. *IOP Conf. Ser. Earth Environ. Sci.*, 704(1), 2021.

[9] M. M. Ahsan et al. Monkeypox diagnosis with interpretable deep learning. *IEEE Access*, 11(July):81965–81980, 2023.

[10] P. Helber, B. Bischke, A. Dengel, and D. Borth. Introducing eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. In *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, page 204–207, 2018.

[11] P. Helber, B. Bischke, A. Dengel, and D. Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, 12(7):2217–2226, 2019.

[12] F. Hu, G.-S. Xia, Z. Wang, L. Zhang, and H. Sun. Unsupervised feature coding on local patch manifold for satellite image scene classification. In *2014 IEEE Geoscience and Remote Sensing Symposium*, page 1273–1276, 2014.

[13] B. Janga, G. P. Asamani, Z. Sun, and N. Cristea. A review of practical ai for remote sensing in earth sciences. *Remote Sensing*, 15(16), 2023.

[14] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[15] V. Narayan, P. K. Mall, A. Alkhayyat, K. Abhishek, S. Kumar, and P. Pandey. Enhance-net: An approach to boost the performance of deep learning model based on real-time medical images. *J. Sensors*, 2023, 2023.

[16] R. O. Ogundokun, S. Misra, A. O. Akinrotimi, and H. Ogul. Mobilenet-svm: A lightweight deep transfer learning model to diagnose bch scans for iomt-based imaging sensors. *Sensors*, 23(2), 2023.

[17] T. Qamar and N. Z. Bawany. Understanding the black-box: towards interpretable and reliable deep learning models. *PeerJ Comput. Sci.*, 9, 2023.

[18] N. Rachburee and W. Punlumjeak. Lotus species classification using transfer learning based on vgg16, resnet152v2, and mobilenetv2. *IAES Int. J. Artif. Intell.*, 11(4):1344–1352, 2022.

[19] V. D. Sachdeva, J. Baber, M. Bakhtyar, I. Ullah, W. Noor, and A. Basit. Performance evaluation of sift and convolutional neural network for image retrieval. 2017.

[20] Ilhem Salah, Khaled Jouini, and Ouajdi Korbaa. Augmentation-based ensemble learning for stance and fake news detection. In *Advances in Computational Collective Intelligence - 14th International Conference, ICCCI 2022, Hammamet, Tunisia, September 28-30, 2022, Proceedings*, volume 1653 of *Communications in Computer and Information Science*, pages 29–41. Springer, 2022.

[21] A. Temenos, N. Temenos, M. Kaselimi, A. Doulamis, and N. Doulamis. Interpretable deep learning framework for land use and land cover classification in remote sensing using shap. *IEEE Geosci. Remote Sens. Lett.*, 20:1–5, 2023.

[22] R. Thakur and P. Panse. Classification performance of land use from multispectral remote sensing images using decision tree, k-nearest neighbor, random forest and support vector machine using eurosat da. *Orig. Res. Pap. Int. J. Intell. Syst. Appl. Eng. IJISAE*, 2022(1s):67–77, 2022. [Online]. Available: https://github.com/phelber/EuroSAT.

[23] Marie Weiss, Frédéric Jacob, and Grégoire Duveiller. Remote sensing for agricultural applications: A meta-review. *Remote Sensing of Environment*, 236:111402, 2020.

.

C.3 Distorted Replicas : Intelligent Replication Schemes to Boost I/O Throughput in NoSQL Systems

Khaled Jouini

14^{th} ACS/IEEE International Conference on Computer Systems and Applications (AICCSA). 2017.

Lien : https://ieeexplore.ieee.org/document/8308258

CORE Rank : C

ICORE Conference Portal

Back to search

ACS/IEEE International Conference on Computer Systems and Applications

Acronym: AICCSA

DBLP Source: https://dblp.uni-trier.de/db/conf/aiccsa

Source: CORE2023

Rank: C

Field Of Research: 46 - Information and Computing Sciences (h-index) (citation)

Source: CORE2021

Rank: C

Field Of Research: 46 - Information and Computing Sciences

Requested as B, retained as C (Data 1) (Decision)

Source: CORE2020

Rank: C

Field Of Research: 46 - Information and Computing Sciences

Source: CORE2018

Rank: C

Field Of Research: 08 - Information and Computing Sciences†

Field Of Research: 12 - Built Environment and Design†

Source: CORE2017

Rank: C

Field Of Research: 08 - Information and Computing Sciences†

Field Of Research: 12 - Built Environment and Design†

Distorted Replicas: Intelligent replication schemes to boost I/O throughput in document-stores

Khaled Jouini
MARS Research Lab LR17ES05
ISITCom, University of Sousse, Tunisia
Email: j.khaled@gmail.com

***Abstract*—NoSQL databases commonly use an aggregate data model, where data that is expected to be accessed together is packed in a single clump and stored in a single node [1]. This aggregate-orientation is essential to running on a cluster as it avoids cross-nodes joins and writes. An important downside of the aggregate data model is that it severely limits the ways data can be efficiently explored and processed, especially as NoSQL systems are being increasingly used by complex data-driven applications, mixing heterogeneous data access patterns.**

Replication is ubiquitous in NoSQL systems. In this work we propose a new replication scheme termed *distorted replicas*. Rather than being physically identical, distorted replicas are logically identical : they restructure replicated data in different ways, but keep the fundamental property of being constructible from one another. By doing so, distorted replicas provide new ways for exploring data, while still ensuring high availability. Experiments conducted in this paper show that even basic distortion schemes allow substantial performance improvements.

***Index Terms*—Aggregate Data Model, Document-Stores, Replication, Physical Design**

I. Introduction

NoSQL systems rise has been driven by the desire to store data on large clusters of commodity servers and to provide horizontal scalability, high availability and high throughput for write/read operations [1]. To ensure high availability, NoSQL systems maintain multiple copies of data on different servers. Most of existing systems implement an *asynchronous replication* which compromises strong consistency in favor of speed and throughput. In this paper, we investigate new forms of replication, termed *distorted replicas*, aiming at speeding up reads and writes in document-stores and aggregate-oriented NoSQL databases.

Document-stores avail the flexibility of structured documents ($e.g.$ JSON) to pack closely related data in a single autonomous document, rather than having them scattered across several tables as in the relational model. By doing so, document-stores manipulate related data in a single database operation and avoid cross-nodes writes and joins, prohibitive in a highly distributed environment. The concept of document is similar to the Domain-Driven-Design (DDD) pattern called *aggregate data model* [2]. Figure 1 depicts a slightly modified JSON-formatted document from the archives of the DBLP bibliography [3].

A key challenge in document-stores is how to model documents in order to meet an application needs in terms of performance and access patterns. Indeed, the whole document-store approach works well only when data access is aligned with the structure of documents [1]. If data is accessed in a different way, the whole system performance may be substantially impacted. Figures 2 and 3 illustrate two possible alternatives to the data model of figure 1. In the data model of figure 2, data is vertically partitioned in three sub-documents, tied up with the same id. In figure 3 data is organized by authors, rather than by publications. The data model of figure 2 is better suited for queries touching a small fraction of fields, as it avoids to load the whole fields in the memory hierarchy. Similarly, the document structure of figure 3 is better suited when data is accessed by authors.While with these simple examples it seems that the data models of figures 2 and 3 outperform the one of figure 1, it is just as easy to come up with examples where the data model of figure 1 is better.

In most cases, we cannot figure out in advance all use cases and data access patterns. Even if we do, it is hard, if not impossible, to find out a document structure capable of efficiently powering all possible queries. Relational databases have an advantage here as they allow to slice and dice data different ways for different queries [1].

Replication is ubiquitous in NoSQL systems. This paper proposes a new form of replication, termed *distorted replicas*, which leverages already existing replication to cope with the heterogeneity of data access patterns. The main idea behind distorted replicas is to restructure replicated data in different ways in order to provide the ability to explore it in different ways. While not physically identical, distorted replicas are logically identical and keep the fundamental property of being constructible from one another. Consequently, distorted replicas help in better supporting various data access patterns, while still ensuring high availability and fault-tolerance.

This paper proposes some basic distorted replication schemes allowing to boost I/O throughput, without additional costs. The proposed strategies are put into practice with MongoDB [4], but are general and applicable to almost all document-stores.

The remainder of this paper is organized as follows. Section 2 describes the main concepts related to document-stores. Section 3 introduces our distorted replicas policies. Section 4 presents related works. Section 4 gives an experimental study of distorted replicas performance. Section 5 concludes the paper.

2161-5330/17 $31.00 © 2017 IEEE
DOI 10.1109/AICCSA.2017.34

```
{"_id": "journals/vldb/KohlerLLZ16",
  "title": "Possible and certain keys for SQL",
  "author": [{"_id": "Henning Köhler"}, {"_id": "Uwe Leck"},
        {"_id": "Sebastian Link"}, {"_id": "Xiaofang Zhou"}],
  "citations": [ ], "pages": "571-596", "year": 2016,
  "url": "db/journals/vldb/vldb25.html#KohlerLLZ16",
  "ee": "http://dx.doi.org/10.1007/s00778-016-0430-9",
  "journal": "VLDB J.", "volume": 25, "number": 4}
```

Fig. 1. JSON-formatted document from the archives of the DBLP bibliography.

```
{"_id": "journals/vldb/KohlerLLZ16","pages": "571-596",
  "title": "Possible and certain keys for SQL",
  "url": "db/journals/vldb/vldb25.html#KohlerLLZ16",
  "ee": "http://dx.doi.org/10.1007/s00778-016-0430-9"
}

{"_id": "journals/vldb/KohlerLLZ16",
  "author": [{"_id": "Henning Köhler"}, {"_id": "Uwe Leck"},
   {"_id": "Sebastian Link"}, {"_id": "Xiaofang Zhou"}],
  "citations": [ ]
}

{"_id": "journals/vldb/KohlerLLZ16", "year": 2016,
  "journal": "VLDB J.", "volume": 25,"number": 4
}
```

Fig. 2. Vertically partitioned document.

II. Anatomy of a document-store

There exists a wide range of academic and commercial document-stores, each with some features that may not exist in others. In the sequel, we use MongoDB as a representative of the feature set, but also reference other document-store systems to discuss features that may be of interest in particular scenarios.

A. Data model design : normalization versus aggregation

Document-oriented databases store and retrieve data in the form of documents formatted in XML, JSON, BSON, and so on. These documents are self-describing, hierarchical tree data structures which consist of field-value pairs. A field value may be a scalar value, an array of scalar values, a document ($i.e.$ embedded document) or an array of embedded documents. Documents with similar structures are typically grouped into *collections* [1]. Collections have no predefined schema and do not enforce document structure [5]. Nonetheless, collections should not be treated as heaps [6]. In practice, documents in a collection ought to share a common subset of fields, in order to comply to at least some common data access pattern and present a consistent and robust data interface to an application [7]. To help in solving this issue, some document-stores such as CouchDB and MongoDB, provide support for checking structural and domain constraints during updates and insertions.

One of the most important document-modeling choices is how to represent relationships between data : with references (normalized data model) or with embedded objects (denormalized data model). References represent the relationships between data by including a link from one document to another, just as in the relational model. With such a normalized model, if related data is stored in separate servers, joins and writes may be prohibitively slow. Embedded documents represent relationships by storing related objects in a single document and hence, avoids cross-nodes joins and writes.

```
{"_id": "Henning Köhler",
"publications": [{"_id": "journals/vldb/KohlerLLZ16",
   "title": "Possible…", "pos": 1,… "author": […],…},
   {"_id": "journals/debu/KohlerLZ16","pos": 1,…},…]}
```

Fig. 3. Data organized according to a different aggregate root ($i.e.$ by authors).

document embedding documents nested inside it, is called in the sequel an *aggregate*. Aggregate is a term that comes from Domain-Driven Design [2] and refers to a cluster of associated objects which are treated as a unit for data manipulation and management. Dealing in aggregates makes it easier to operate on a cluster, since aggregates make natural units for sharding and replication [1].

B. Replication

In clusters of hundreds, or even thousands, of geo-distributed commodity servers, high rates of hardware and network failures are common and the system should be able to instantly cope with them [7]. Replication is the process of maintaining different replicas of the same data on different servers. The primary purpose of replication is to enhance availability and fault-tolerance by providing multiple paths to redundant data. Replication can also be used to increase : (i) I/O throughput by distributing requests across servers; and (ii) data locality by allowing a client application to access data from the closest server.

A set of servers maintaining replicas of the same data (sub-)set is called a *Replica set* in MongoDB. A replica set is composed by one master node, called *primary*, a set of slave nodes, called *secondaries* and one optional node called *arbiter*. The primary node is the only member in a replica set that receives writes. When the primary receives a write request, it updates its data set and records the write in a special collection, called the *opLog* ($i.e.$ operations Log). Secondary nodes periodically import the opLog from the primary or from any other member of the replica set. Secondaries then apply all changes to their local replicated collections in such a way that they reflect the master collections [4]. In case of a master failure, an eligible slave holds an election to elect itself the new master [4]. When the node that failed comes back, it joins in as a slave and synchronizes its local collections. An arbiter server does not replicate the master collections. Its only purpose is to obtain a quorum during the election of a new master.

As in most NoSQL systems, replication in MongoDB is by default asynchronous : (i) there may exist a delay between the occurrence of an operation on the primary and its application on a secondary ($i.e.$ *replication lag*); and (ii) the client application does not have to wait for the completion of a write on slaves. To ensure strong consistency, reads are by default sent to the primary and hence only access the most recent data copy. In such case, replication is only used for availability and fault-tolerance. If it is required to use replication for increased read throughput, a client application may choose to read from one of the existing slaves, with the risk of reading outdated data.

In contrast to MongoDB, some systems such as CouchDB and Amazon Dynamo allow a master-master replication, in which writes and reads are distributed over replicas. Distributing writes prevents from having a single server supporting all writes, but raises consistency issues, as separate servers may create conflicting versions [7].

C. Sharding

In most NoSQL systems, replication is not the primary mean for scaling and sharding is often a better strategy. Sharding is similar to horizontal partitioning in RDBMSs. It consists in splitting a data set according to a given field, called the *shard key*. The resulting data subsets are called $chunks$ and are hosted on multiple separate servers, called *shards*. Each shard is an independent database having its own subset of data stored on its own local disks [4]. Typically, NoSQL systems do not support distributed joins and implement limited distributed transactions capabilities, due to the I/O overhead and coordination required [8].

A prominent concern in sharding is to balance the load between shards. Typically, when a chunk grows beyond a given size, it is split causing an increase in the number of chunks held by the server. If the chunk distribution becomes uneven, some chunks are migrated from the shard that has the largest number of chunks to the one with the least number of chunks, until the cluster is rebalanced. A similar process occurs when a new server is added to the cluster.

The mapping between chunks and shards ($i.e.$ mapping metadata) need to be kept to be able to route subsequent read and write requests to the appropriate shard. Some shared-nothing architectured systems, such as Amazon Dynamo, duplicate some of the mapping metadata on each shard of the cluster. Other systems such as MongoDB store mapping metadata in a separate centralized server ($i.e.$ the *Config Server*).

In MongoDB each shard can be a complete replica set ($i.e.$ data is first distributed and then replicated). In addition to shards and config servers, a MongoDB sharded cluster is composed by a set of query routers, named mongos. When a mongos instance starts, it loads a copy of the config server database and route the reads and writes from client applications to shards.

III. Distorted replicas

Distorted replicas leverages already existing replication to efficiently solve the problem of access patterns heterogeneity. With distorted replicas, instead of having a document and its replicas physically identical, they are only logically identical. By logically identical we mean that data is organized differently in each replica, but the different replicas can be constructed from one another.

This section exposes the main consequences of the aggregation model and explores different basic schemes for solving them with distorted replicas. In the sequel we adopt the MongoDB architecture for its flexibility, but similar schemes can be implemented with other document-stores.

A. Vertical partitioning

1) Write/read overheads: Aggregates are useful in that they pack into one document, objects that are expected to be accessed together. However, there are many use cases where objects or fields need to be accessed individually. Consider for instance the relationship between a publication and its authors. Some applications will want to read the authors emails whenever they access a publication; this fits in well with combining the publication with its authors into a single aggregate. Other applications, however, want to access the email of an author and thus only access the author's data. When a field needs to be accessed individually, not only that field's value is loaded in the memory hierarchy, but also all the data within the same aggregate. As aggregates are commonly large in size, loading large amount of data irrelevant for a given query may seriously wastes main memory and disk bandwidths and increases the amount of CPU cycles wasted in waiting for data loading [9]. The introduced write/read overhead is one of the most important downsides of the aggregate approach.

A more subtle limitation is that in some cases the update or the insertion of a field in a document, may lead to the entire document re-write. This happens when the update or the insertion of new fields causes a growth in the document size beyond the allocated space for that document. If space is pre-allocated for each document, the fragmentation of data files is avoided, but even with pre-allocation, updating documents gets slower as documents grow.

2) Principle: The aim of the first scheme of distorted replicas is to store a collection replica at a lower level of granularity to reduce the write/read overheads. It consists in partitioning vertically a base document with n fields in $m \leq n$ sub-documents, with the assumption that data of a particular sub-document will usually be accessed together. As illustrated in figure 2, each sub-document holds : (i) one or more fields; and (ii) the base document id identifying the original base document that the values came from. Vertical partitioning of a collection of documents, results in sub-collections which are similar to column families in wide-column stores ($e.g.$ Cassandra and Hbase). As in wide-column stores, subsets of fields are stored together and the different pieces of the same row ($i.e.$ document) are identified by the same id.

Vertical partitioning has poor space utilization as it stores the base document id along with each of the vertical partitions [10]. Additional space may also be lost as each partition is padded and stored with its own header. Vertical partitioning performs also poorly for insertions because multiple distinct (sub-) documents have to be inserted for each inserted base document.

Despite its drawbacks, vertical partitioning has one great benefit against the aggregate model : a query can only load the sub-documents that it requires, instead of loading the whole base document. This is especially helpful for workloads with high rate of updates or reads with low projectivity ($i.e.$ touching a small fraction of fields).

We should notice that the idea of maintaining two copies of the same data, one stored following the Decomposition

Storage Model [11] and the other following the N-ary Storage Model has been first introduced in [12] and applied to RDBMS running on servers with mirrored disks ($i.e$ RAID 1). [12] proposes many schemes to speed up joins of sub-relations. In our work we further extend the approach to document-stores running on a cluster of replicated servers. We are less interested by joining sub-documents, as in our case, joins should be performed only when we need to reconstruct the base replica from the vertically partitioned one. Operations accessing data from different sub-collections, have to be directed to the replica hosting the aggregated documents.

3) Sub-documents storage: Sub-documents resulting from vertical partitioning can either be stored in the same collection or in separate collections. If the sub-documents are put in the same collection, they cannot be identified with the id of the base document. The id must nonetheless be added to each of the sub-documents to allow the reconstruction of the base document. The id can also acts as the collection shard key to ensure that all the partitions originating from the same document fall within the same shard. The main strength of this scheme is that it allows to store the sub-documents sorted according to the document id. Having the sub-documents stored contiguously on the disk allows to efficiently access all the partitions of the same document and can be an alternative to joins if the base document needs to be reconstructed.

The main weakness of this scheme is that queries with low projectivity are constrained to access all the sub-documents and to perform a type test for each accessed one. Consider the example of figure 2 and a map/reduce job searching for the publications of each author. If the sub-documents are placed in the same collection, the map is forced to treat each sub-document, and, for each, to test whether or not it contains an "author" field.

Putting sub-documents in separate collections allows to only process sub-documents relevant for a given query and avoids type tests. The counterpart of this strategy is that it requires to loop over more than one cursor if more than one partition need to be scanned. Another important limitation is that in some systems, such as MongoDB, a map reduce job can only be processed over a single collection. As a consequence, performing a map reduce job over fields that belong to different sub-collections, may be tedious. In our opinion, putting the sub-documents in separate collections is better than storing them in the same one, as it avoids loading unnecessary data in the memory hierarchy. If a query or a map/reduce job needs to read data from more than one sub-document, it can be directed to the replica with the aggregate model.

B. Multiple aggregate roots

Objects in an aggregate are bound together by a root object, known as the *aggregate root* [2]. In most cases, there exist many root candidates. Using our DBLP example, publications, authors and conferences are all root candidates. Bounding objects by one of the candidate root may help with some data interactions, but is necessarily an obstacle for many others. As reported in [2], the whole aggregate orientation approach works well only when data access is aligned with document roots.

The idea behind the "Multiple Aggregate Roots" scheme is to use different aggregate roots for a collection and its replicas. Using our DBLP example, the base document collection may be rooted at publications, while one of its distorted replicas may be rooted at authors (see figure 3). This can be easily implemented by processing the database log (see section V).

If documents are rooted at publications, distorted replicas prevent from digging in every document to find out an author publications. Nonetheless, if the collection is sharded, distorted replicas do not prevent from visiting each shard. The number of accessed shards can be reduced by maintaining a Counting Bloom Filter (CBF) [13], [14] for each shard and host the filters within the metadata server ($e.g.$ the Config Server in MongoDB). As a regular Bloom Filter (BF), a CBF allows to quickly test whether an author appears in a given shard. Unlike a BF which associates a single bit to each of the filter buckets, a CBF associates a counter with a fixed size of c bits. The i^{th} counter indicates the number of elements currently hashed to the i^{th} bucket.

Our choice of using CBFs instead of BFs is motivated by data movements between shards, unavoidable to maintain a balanced data distribution. If documents from a replicated collection are migrated from one shard to another, the filters associated with the two shards, may need to be updated accordingly. Let consider the DBLP dataset (where data is organized by publications) and a distorted replica where data is organized by authors. Suppose that p is a publication that need to be migrated from a shard s_i to a shard s_j, and a one of its authors. Moving p results in : (i) the addition of a to the filter associated to s_j; and (ii) the deletion of a from the filter associated to s_i, if p were the only publication of a in s_i. While a BF does not allow deletions of elements, deleting an element from a CBF can be safely done by decrementing the relevant counters.

IV. Related work

The problem of heterogeneity in data access patterns has been addressed in different ways. This section focuses on approaches that use different physical designs for replicas of the same data collection.

A. Secondary indexes and the divergent physical design

In [15] the authors propose a novel tuning paradigm for replicated databases, coined *divergent designs*. Given a replicated database, a divergent design indexes the same data differently in each replica, and hence, specializes replicas for different subsets of the workload [15]. With this design, each query is routed to the replica that can evaluate it most efficiently. The idea of divergent design was further developed in [16] where the authors proposed RITA, an index-tuning advisor for replicated databases. RITA allows to : (1) generate fault-tolerant divergent designs; and (2) spread evenly the load over replicas.

As aptly stated in [1] with secondary indexes we're still "working against the aggregate structure". Indeed, by indexing documents on author identifiers, some of the latency would be hidden, but performance would only be sub-optimal, since the retrieved documents are typically scattered across the storage device. Secondary indexes are also not helpful for map/reduce jobs and for queries involving regular expressions. In some cases, scanning the whole collection is a better solution. Using our DBLP example, as authors may have variations in their first names ($e.g.$ "Mike Stonebraker", "Michael StoneBraker", etc.), it is common to use a regular expression to find out their publications. With the DBLP dataset and the MongoDB settings of section V, a query looking for the publications of any author having Stonebreaker as last name (authors._id: {Stonebraker$}) takes on average 306916 ms with a secondary index on authors ids, 34776 ms without an index and 8742 ms if data were organized by authors.

B. Divergent block layouts

[17] proposes Trojan Layout, a data layout inspired by PAX (Partition Attributes Across) [18] and intended to improve data access times in Hadoop Distributed File System (HDFS). Given a relation R with arity n, PAX partitions each block into n miniblocks. The i^{th} miniblock stores all the values of the i^{th} attribute of R. The Trojan layout splits a HDFS block into $m \leq n$ miniblocks and stores in each miniblock the values of k ($1 \leq k \leq$) attributes ($i.e.$ vertical partitioning inside each chunk). Trojan Layout provides a high degree of spatial locality when the values of the k grouped attributes are sequentially accessed and avoids to read $(n-k)$ irrelevant attributes for a given query. To better handle a mix of queries with different access patterns, [17] proposes also to group attributes differently in each HDFS block (chunk) replica according to the query workload.

The Trojan Layout is helpful for queries that sequentially access some given attribute values (of a flat records), but it can't help with queries that need to access data according to a different aggregate root. As for secondary indexes, with the Trojan Layout we're still "working against the aggregate structure". Using our DBLP example, the Trojan Layout allows to group author values at the chunk level. However, it does not allow to group publications by authors at the shard level (which require a complete restructuration of the data). Thus, with the Trojan Layout, a query such as "find Stonebreaker's publications" will always require to perform a map/reduce Job and to process every single publication in the collection. With the multiple aggregate roots model the query only requires to visit one document in each shard where the author appears.

Note here that the multiple aggregate roots model and the Trojan Layout are not antagonist: one can use the first model to restructure data according to a different aggregate root and then store the restructured data with the Trojan layout to further improve queries that sequentially access some given attribute values.

C. Materialized views

In RDBMSs, a materialized view (MV) is a named query whose results are persisted. The primary purpose of materialized views is to optimize an expensive read query by precomputing and caching all or part of its intermediate results.

NoSQL systems, such as CouchDB, allow the creation of MVs (they are called views, but are more akin to MVs since their results are persisted). As in most NoSQL systems, a view in CouchDB is a persisted result of a map-reduce computation. This result takes the form of a new collection organized as a B-tree built on the id [7]. MongoDB provides a similar feature through the so called "incremental map-reduce". As CouchDB views, incremental map-reduce consists in persisting the result of a map-reduce job in a new collection that can be queried, replicated and sharded as a regular collection.

There are two rough strategies to refresh an MV: eager or lazy. With the eager approach, the MV is updated at the same time as the base data. This approach allows to keep MVs as fresh as possible. However, it may lead to an important write overhead, as each write operation needs to be instantly propagated not only to the document replicas, but also to all the MVs derived from the updated collection.

The lazy approach adopted by CouchDB, consists in refreshing an MV when it is queried and by re-evaluating the map-reduce only on changed documents [19]. Processing updates by batch at query time, reduces the write overhead, but may significantly slowdown reads. An alternate refresh scheme is adopted by MongoDB, where an incremental map-reduce is re-evaluated on demand. Such an approach may lead to situations where client applications read stale data.

As materialized views, distorted replicas allow to restructure documents to provide new ways for exploring data. The main difference is that distorted replicas take advantage of the already existing replication to generate restructured data without any additional refresh cost, while MVs introduce a significant write overhead. Whether handled at the base data update time or shifted to the view query time, the write overhead may substantially impact performance.

It is worth noticing that if MVs were enabled to regenerate base data in a two-way replication scenario, they would be an interesting tool to implement distorted replicas. Such a scenario is permitted in some RDBMSs such as Oracle which uses MVs to locally replicate remote data in a replication environment. Oracle allows in addition to create "updatable materialized views" which enable to insert/update/delete data through an MV.

D. Pluggable storage engines

Some database systems, such as MySQL, Dynamo and MongoDB are shipped with different storage engines and applications are allowed to choose the one that is the most appropriate to their workloads. One of the most interesting features of MongoDB is that it allows the use of multiple storage engines within a single replica set ($i.e.$ the replicas of the same document can be stored and processed by different

storage engines). The pluggable storage engines approach can be considered as such as a special case of distorted replication.

A shortcoming of the above approach is that document replicas remain physically identical ($i.e.$ have the same structure). Even with the most suitable storage engine, if a structure is not optimal for a given access pattern, performance can only be sub-optimal : the storage engine can at most hide some of the latency, but can not hide it all.

V. Performance Evaluation

Most of existing NoSQL benchmarks, such as [20], focus on NoSQL systems performance in terms of throughput and latency. They typically treat data as heaps of key-value pairs where values are opaque : queries are often limited to keys, read operations are assumed to retrieve the whole record fields and the update of a set of fields or of a single field are not distinguished.

In this work we are interested in bringing to light the potential benefits expected from restructuring the replicas of the same data in different ways. Consequently, we need to capture data semantics to identify root candidates and to perform vertical partitioning. To quantify the potential benefits of each distortion scheme, we used therefore a real dataset based on the DBLP Computer Science Bibliography [3] and a set of realistic workloads.

In the remainder, subsection V-A presents the tools used for synchronizing the proposed distorted replicas. Subsection V-B describes the main features of the DBLP data set. Subsection V-C defines the workloads considered and discusses the results.

A. Prototype for distorted replicas synchronisation

Whether distorted or not, replicas have to be kept synchronised when data is inserted, updated or deleted. In our work, we implemented a distorted replicas synchronizer based on the operations log and tailable cursors.

MongoDB keeps track of all operations that change the state of replicated documents in a special collection, called the opLog, stored in a special database called local (`Local.opLog.rs`). Each document in the opLog corresponds to a single operation performed on the primary node. OpLog documents contain several fields, including [21] : the timestamp when an operation was performed, the type of the operation performed, the name of the collection affected by the operation and the new state of the affected document.

As the write ahead log of most of RDBMSs, the opLog has a fixed size ($i.e.$ capped collection) and behaves like a circular queue : when it reaches its max size, the newest entries replace the oldest ones. The opLog collection can be queried just as a regular collection, from the Mongo shell or from any MongoDB driver. To synchronise our replicas without replaying already applied operations, it is necessary to only see the latest entries added to the opLog. MongoDB doesn't have triggers and automatically close a cursor when it exhausts its result set. To prevent from requerying over and over the opLog to get newly added entries, MongoDB uses and allows to define a special kind of cursors, called *tailable cursors*. A tailable cursor is a kind of listener that only shows new data as it is written to a collection. When a tailable cursor reaches the end of the result set, it is not closed. Rather, it sleeps and waits for more documents to be added to the collection. If additional documents are inserted, the tailable cursor retrieves them. Tailable cursors are the natural way to intercept opLog's new entries (MongoDB uses tailable cursors to tail the opLog [4]).

B. Dataset and settings

The DBLP Computer Science Bibliography, is a data set containing bibliographic information on scientific publications. All the DBLP records are distributed in one big XML file. Each record is associated with a set of fields representing bibliographic data relevant with respect to its type and has an id field that uniquely identifies it. Ids resembles to slash separated Unix file names [3].

We developed a DBLP parser in Java following the recommendations of [3]. Currently our parser only extracts "article" and "inproceeding" records. The extracted records were inserted in the same collection. The resulting MongoDB collection contains about 3.1 million publication documents embedding about 1.6 million authors. The average document size is 538 bytes and the whole collection size is 1.5 GB.

Two distorted replicas were dynamically generated using the synchronizer discussed in subsection V-A. The first one uses vertical partitioning as discussed in subsection III-A. This storage model is referred to as VPM in the sequel. With this scheme each base document is partitioned in three subdocuments as illustrated in figure 2. The average object size in each of the three sub-collections is respectively 254 kB, 191 kB and 102 kB.

The second distorted replica organizes data by authors as discussed in subsection III-B. We added a pos field to each author in order to keep the order in which an author appears in a given publication. The resulting distorted replica is composed by 1.6 million documents having an average size of 2.3 MB. In the sequel, the aggregate data model where data is organized by pubications is referred to as ADM1 and the one where data is organized by authors as ADM2.

Simulations were performed on a dedicated dual core i5-3230M system, running Ubuntu 16.04.1 LTS. Each core offers a base speed of 2.6 GHz and the two cores can handle up to four simultaneous threads. This computer features 4 GB main memory (DDR3-1600MHz), 128 kB L1 cache, 512 kB L2 cache and 3 MB L3 cache. The hard disk is a Serial-ATA/600 having a rotational speed of 7200 rpm. Experiments were conducted on MongoDB release 3.2 and its storage engine WiredTiger. MongoDB was run using its default settings and no special tuning was done. As the Mongo shell does not allow to manipulate simultaneously documents from different collections, we conducted our tests through the Java driver. All reported times are the average of three consecutive runs.

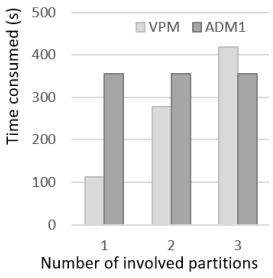


Fig. 4. Scan cost as function of projectivity.

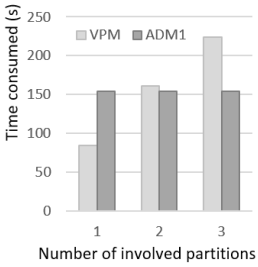


Fig. 5. Query cost as function of projectivity.

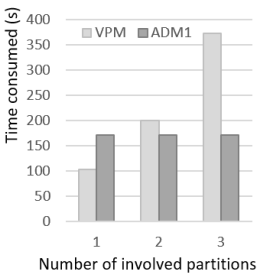


Fig. 6. Update cost as function of projectivity.

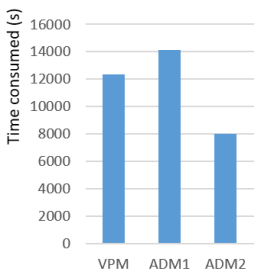


Fig. 7. Average time consumed when searching for the co-authors graph.

C. Workload and results

1) Relevancy of vertical partitioning:

a) Insertion Cost: The insertion cost is measured by the average time elapsed during the insertion of 3.1 millions documents. It equals 1714 s with ADM1 and 3854 s with VPM. As expected, insertions are faster with ADM1. This is due to the fact that a single write suffices to push all the fields of a document, while with VPM, the server handles write requests for multiple collections which are likely scattered across the disk. The performance of VPM should be worst if the base collection were decomposed in more than 3 partitions.

b) Scan cost: Figure 4 depicts the average time consumed by a full collection scan as function of projectivity (*i.e.* number of involved partitions). The only case where ADM1 outperforms VPM is when all sub-collections need to be scanned.

c) Query cost: To evaluate the query cost, we considered three query types. The first type represents queries that only touch fields belonging to the same sub-collection. The second type (resp. the third type) represents queries that requests fields belonging to 2 sub-collections (resp. 3 sub-collections). Type 2 queries are equivalent to joins over 2 sub-collections. Type 3 queries are equivalent to the reconstruction of base documents from their sub-partitions. For each query type, 100K publication `_ids` were randomly selected using the `$sample` operator of MongoDB. For each `_id`, we measure the average time needed to extract the corresponding fields, from one sub-collection, two sub-collections and three sub-collections. The obtained results are compared to the time consumed to extract the same fields from the base collection.

As shown in figure 5, ADM1 performance are stable. This is due to the fact that the whole document need to be extracted, regardless the number of requested fields. For type 1 queries, VPM consumes on average 48% less time than the aggregate model. For type 2 queries, ADM1 performance are equivalent to VPM. For type 3 queries ADM1 is 38% faster than VPM.

d) Update cost: As for the query cost, to evaluate the update cost we consider 100K randomly selected publication `_ids` and three update types. Updates of type 1 (resp. type 2 and 3) involve scalar fields belonging to one partition (resp. 2 and 3 partitions). The considered updates do not grow the document size. When the updated fields belong to one partition, VPM is 40% faster than ADM1. As shown in figure 6, VPM performance deteriorates if fields from more than one partition are updated at a time. In our experiments, ADM1 is 46% faster than VPM when 3 partitions are involved. We should notice here that in practice updating all, or almost all of the fields of a record at a time is unusual. In such case, a delete followed by an insert provides generally better results.

e) Map/Reduce: To illustrate the behavior of VPM and ADM1 with regard to map reduce jobs, we consider the classical case consisting in finding the co-authors graph : for each pair of authors who have at least collaborate on a publication, we calculate the total number of publications they participated in. As expected VPM outperforms ADM1, as it only loads the author field, while ADM1 loads all the fields into the memory hierarchy.

2) Relevancy of the multiple aggregate roots scheme: To give an order of magnitude of the gains that could be drawn from structuring data with different aggregate roots, we considered 5 realistic queries/map-reduce jobs, generally simple to process with one replica, but tedious with the other.

a) Query 1: "Find the publications of each author": If data is organized by authors, a simple `find()` from the mongo shell or the Java driver would be sufficient to get the publications of each author (less than 1 ms). However, when data is organized by publications, it is necessary to perform the map/reduce job, consuming on average 8155 s (from the Mongo shell).

b) Query 2: "Find the authors of each publication": Inversely to Query 1, a simple find operation is sufficient (less than 1 ms) if data is organized by publications and a map/reduce job is necessary if data is organized by authors. The average time required to process the map/reduce job is 7873 s (from the Mongo shell).

c) Query 3: "Find the publications of a given author": In our parsed DBLP dataset, the average number of publications per author is 5.46. We randomly selected 3 authors having each 6 publications. When data is organized by authors, such a query consumes less than 1 ms. When it is organized by publications it consumes on average 29 s (from the Mongo shell).

d) Query 4: "Find the authors of a given publication": The average number of authors per publication in our parsed DBLP dataset is 2.85. We randomly selected 3 publications co-written by 3 authors. If data is organized by publications,

finding the authors of a given publication is executed in less than 1 ms. When it is organized by authors, it requires on average 25 s (from the Mongo shell).

e) Query 5: "Find the co-authors graph": As illustrated in figure 7 when data is organized by authors, the map/reduce job consumes on average 35% less execution time than when it is organized by publications.

f) Sharded setup: In this set of experiments, we considered a very simple sharded setup, where the documents of the collection publications are distributed over two mongod instances, running on the same machine. Each of the two shards holds a sub-set of the base documents and the corresponding distorted replica, containing the same data organized by authors. The first shard contains 1.769.638 publications and their corresponding 1.059.357 authors. The second shard contains 1.346.159 publications and 988.934 authors. We considered a simple query searching for the publications of each author in the data set. A Counting Bloom Filter (CBF) is associated to each shard in order to indicate which authors appear in which shards. Without CBF, a query is systematically sent to both shards, and the returned results are merged. With CBF, the query is only sent to the shard where a given author appears. The average execution time is 2387 s with CBF and 2983 s without (from the Java driver). Even with our simplistic shard setup, CBF help in reducing the query execution time by mean of 20%. We should notice here that the gain allowed by CBF would be higher if the shards run on different machines, as unnecessary requests need to be transferred across the network.

VI. Conclusion

Aggregates are useful in that they pack into one document, data that is expected to be accessed together. Aggregates are essential to running on a cluster, but severely limit the ways data can be efficiently explored and processed.

This paper introduced distorted replicas, a new replication scheme helping in better handling data access heterogeneity in document-stores. Distorted replicas take advantage of already existing replication to restructure the same data in different ways. By doing so, distorted replicas provide new ways for exploring data, while still preserving the ability to reconstruct replicas from one another.

The paper studied basic distortion schemes. In the first one, replicated data is vertically partitioned. In the second one, replicated data is organized according to a different aggregate root. Based on the workload mix, the complexity of queries, and the update frequency, an application can direct its reads and writes to the distorted replica that is most suitable. We showed that even with simplistic schemes, performance can be substantially improved.

The main goal of this work was to bring to light the relevancy of distorted replicas and to motivate research in this direction. We implemented for this purpose a prototype for synchronizing distorted replicas, based on the operations log available in most NosSQL systems. This prototype is sufficient as a proof of concept, but many challenging problems need to be thoroughly thought out. As a part of our future work, we intend to further examine the dynamic migration of data between shards and its impact on distorted replicas and on load balancing. This especially holds when data and its replica are arranged according to different aggregate roots. In this paper we recommend the use of counting bloom filters to ease data movements between shards, but more sophisticated schemes can be implemented. Another interesting point to consider is the definition of a cost model for each type of data organization, to help query optimizers in determining the replica to which a query should be directed.

References

[1] P. J. Sadalage and M. Fowler, *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence*, 1st ed. Addison-Wesley Professional, 2012.

[2] E. Eric, *Domain-Driven Design: Tacking Complexity In the Heart of Software*. Addison-Wesley Longman Publishing Co., Inc., 2003.

[3] M. Ley, "DBLP – Some Lessons Learned," *PVLDB*, vol. 2, no. 2, pp. 1493–1500, 2009.

[4] "Mongodb," http://www.mongodb.com, [Accessed: 2016-09-01].

[5] R. Arora and R. R. Aggarwal, "Modeling and Querying Data in MongoDB," *International Journal of Scientific and Engineering Research*, vol. 4, no. 7, pp. 141–144, 2013.

[6] L. Wang, S. Zhang, J. Shi, L. Jiao, O. Hassanzadeh, J. Zou, and C. Wangz, "Schema Management for Document Stores," *Proc. VLDB Endow.*, vol. 8, no. 9, pp. 922–933, May 2015.

[7] S. Abiteboul, I. Manolescu, P. Rigaux, M.-C. Rousset, and P. Senellart, *Web Data Management*. Cambridge University Press, 2011.

[8] "Oracle nosql database concepts manual," http://docs.oracle.com/cd/NOSQL/html/ConceptsManual/index.html, [Accessed: 2016-09-01].

[9] K. Jouini, G. Jomier, and P. Kabore, "Read-optimized, cache-conscious, page layouts for temporal relational data," in *Proc. of the 19th Int. Conf. on Database and Expert Systems Applications*, ser. DEXA '08, 2008, pp. 581–595.

[10] K. Jouini and G. Jomier, "Modèles de stockage orientés interrogation pour bases de données temporelles," *Ingénierie des Systèmes d'Information*, vol. 15, no. 1, pp. 61–85, 2010.

[11] G. P. Copeland and S. Khoshafian, "A Decomposition Storage Model," in *ACM SIGMOD'85*. ACM Press, 1985, pp. 268–279.

[12] R. Ramamurthy, D. J. DeWitt, and Q. Su, "A Case for Fractured Mirrors," *The VLDB Journal*, vol. 12, no. 2, pp. 89–101, 2003.

[13] L. Fan, P. Cao, J. Almeida, and A. Z. Broder, "Summary Cache: A Scalable Wide-area Web Cache Sharing Protocol," *IEEE/ACM Trans. Netw.*, vol. 8, no. 3, pp. 281–293, Jun. 2000.

[14] B. Fan, D. G. Andersen, M. Kaminsky, and M. D. Mitzenmacher, "Cuckoo Filter: Practically Better Than Bloom," in *ACM Int. Conf. on Emerging Networking Experiments and Technologies*, ser. CoNEXT '14. ACM, 2014, pp. 75–88.

[15] M. P. Consens, K. Ioannidou, J. LeFevre, and N. Polyzotis, "Divergent physical design tuning for replicated databases," in *ACM SIGMOD Int. Conf. on Management of Data*, ser. SIGMOD '12, 2012, pp. 49–60.

[16] Q. T. Tran, I. Jimenez, R. Wang, N. Polyzotis, and A. Ailamaki, "Rita: An index-tuning advisor for replicated databases," in *Proc. of the Int. Conf. on Scientific and Statistical DB Mgmt*, ser. SSDBM '15, 2015, pp. 22:1–22:12.

[17] A. Jindal, J.-A. Quiané-Ruiz, and J. Dittrich, "Trojan data layouts: Right shoes for a running elephant," in *Proc. of the 2Nd ACM Symposium on Cloud Computing*, ser. SOCC '11, 2011, pp. 21:1–21:14.

[18] A. Ailamaki, D. J. DeWitt, M. D. Hill, and M. Skounakis, "Weaving relations for cache performance," in *VLDB'01*. Morgan Kaufmann Publishers Inc., 2001, pp. 169–180.

[19] P. Agrawal, A. Silberstein, B. F. Cooper, U. Srivastava, and R. Ramakrishnan, "Asynchronous View Maintenance for VLSD Databases," in *ACM SIGMOD'09*. ACM, 2009, pp. 179–192.

[20] B. F. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan, and R. Sears, "Benchmarking Cloud Serving Systems with YCSB," in *ACM Symposium on Cloud Computing*, ser. SoCC '10. ACM, 2010, pp. 143–154.

[21] K. Chodorow and M. Dirolf, *MongoDB: The Definitive Guide*, 1st ed. O'Reilly Media, Inc., 2010.

.

C.4 Fast and Accurate Fingerprint Matching using Expanded Delaunay Triangulation

Mohamed Hédi Ghaddab, Khaled Jouini & Ouajdi Korbaa

14^{th} ACS/IEEE International Conference on Computer Systems and Applications (AICCSA). 2017.

Lien : https://ieeexplore.ieee.org/document/8308364

CORE Rank : C

ICORE Conference Portal

Back to search

ACS/IEEE International Conference on Computer Systems and Applications

Acronym: AICCSA

DBLP Source: https://dblp.uni-trier.de/db/conf/aiccsa

Source: CORE2023

Rank: C

Field Of Research: 46 - Information and Computing Sciences (h-index) (citation)

Source: CORE2021

Rank: C

Field Of Research: 46 - Information and Computing Sciences

Requested as B, retained as C (Data 1) (Decision)

Source: CORE2020

Rank: C

Field Of Research: 46 - Information and Computing Sciences

Source: CORE2018

Rank: C

Field Of Research: 08 - Information and Computing Sciences†

Field Of Research: 12 - Built Environment and Design†

Source: CORE2017

Rank: C

Field Of Research: 08 - Information and Computing Sciences†

Field Of Research: 12 - Built Environment and Design†

2017 IEEE/ACS 14th International Conference on Computer Systems and Applications

Fast and Accurate Fingerprint Matching using Expanded Delaunay Triangulation

Mohamed Hedi Ghaddab
MARS Research Lab LR17ES05
ISITCom, University of Sousse, Tunisia
Email:ghaddab.mohamedhedi@gmail.com

Khaled Jouini
MARS Research Lab LR17ES05
ISITCom, University of Sousse, Tunisia
Email: j.khaled@gmail.com

Ouajdi Korbaa
MARS Research Lab LR17ES05
ISITCom, University of Sousse, Tunisia
Email: Ouajdi.Korbaa@centraliens-lille.org

***Abstract*—Efficient and reliable fingerprint matching is crucial for many civilian and forensic applications. Fingerprints are characterized by large intra-class variations ($i.e.$ variability in different impressions from the same finger). This variability manifests itself by the missing or the displacement of genuine minutiae and the detection of spurious minutiae. Displaced, missing and spurious minutiae make fingerprint matching a very challenging pattern recognition problem.**

Although significant improvements have been made in minutiae-based matching, most of minutiae matching algorithms lack of robustness with respect to displaced, missing and spurious minutiae [4]. This paper introduces EDT-C, a new fingerprint matcher based on minutiae triplets. The proposed matcher uses an extended form of Delaunay Triangulation that allows to take into account, not only genuine minutiae, but also spurious, displaced and missing ones. EDT-C characterizes minutiae triplets by a set of innovative geometric features that help in tolerating linear and non-linear distortions. Finally, EDT-C includes some optimizations that allows to quickly consolidate local matchings and filter non-matching minutia triplets. Experiments show that EDT-C has a reasonable computational cost and is far more accurate than its main competitors.

***Keywords*—Minutiae-based fingerprint matching; Expanded Delaunay Triangulation; Triplet descriptor**

I. Introduction

Fingerprint-based recognition systems are being extensively used in a broad range of civilian and forensic applications [26]. Fingerprints are used either for verification or identification. The purpose of verification is to corroborate the identity claimed by a person by comparing its captured fingerprint(s) with her own pre-stored template ($i.e.$ one-to-one comparison). The purpose of identification is to establish the identity of a person, given a query impression I and a database of fingerprint templates ($i.e.$ one-to-many comparisons). Fingerprint matching is a crucial step in both verification and identification problems [12].

Minutiae are the points of a fingerprint where its ridge lines end or bifurcate. Due to their high distinctiveness, minutiae-based fingerprint matching is one of the most widely accepted recognition technology [7]. In this work we are only interested in minutiae-based fingerprint matching. Given two sets of minutia points, the aim of minutiae-based matching algorithms is to find the alignment that maximizes the number of matching minutiae and to derive a similarity score accordingly [20].

Several causes make minutiae-based matching a very challenging problem: linear distortions ($e.g.$ translations, scale, rotation, etc.), non-linear distortions (resulting from mapping the three-dimensional shape of a finger onto a two-dimensional image) and other data source damages such as scars, sweat, small overlap and dryness [18]. These factors lead to large intra-class variations ($i.e.$ variability in different impressions from the same finger), which manifests itself by the detection of spurious minutiae and the missing or the displacement/disorientation of genuine minutiae [17].

Minutiae matching can be classified as local and global. Global minutiae matching tries to simultaneously superpose the whole minutia points of a fingerprint I to their mates in a fingerprint T. Local minutiae matching compares I and T according to local structures ($i.e.$ local regions). A local structure is typically formed by a minutia p and minutiae lying within p neighborhood.

Local structures are commonly described by attributes ($e.g.$ angles, distances, etc.) invariant with regard to linear distortion ($e.g.$ translation, rotation, etc.) and can therefore be matched without any prior global alignment. Local structures require also less computational resources than global matching and are more robust to non-linear distortions and partial overlaps [5]. However, by relaxing global spatial relationships, local structures reduce the amount of information available for discriminating fingerprints [17].

To retain the advantages of the global and the local matchings, most of recent minutiae-based algorithms perform a local matching followed by a global matching [20]. The local structures matching quickly and reliably determine pairs of local structures from I and T that match. The global matching step, called the *consolidation step*, checks whether the local matches are consistent at the global level [12].

Many approaches have been proposed to construct local structures. This includes minutiae cylinder [4], texture mixed [21] and minutiae triangles [25], [14], [23], [18]. A minutiae triangle ($i.e.$ triplet) is a structure formed by three minutiae points. Minutiae triangles ($i.e.$ triplets) have the following advantages with regard to other local structures [18]: low computational complexity, tolerance to fingerprint deformations, high discriminative power, embeddability on light architectures, compliancy with interoperability standards (most popular standards are based only on minutiae), etc. In the sequel we only focus on approaches based on minutiae triangles.

Although significant improvement has been made in triplets-based matching, state-of-the-art algorithms lack of

2161-5330/17 $31.00 © 2017 IEEE
DOI 10.1109/AICCSA.2017.33

robustness with regard to displaced, missing and spurious minutiae [4].

In this paper we introduce a novel triplets-based fingerprint matcher, called EDT-C, designed to be robust with respect to missing, spurious and displaced minutiae, and to ensure reasonable computational costs. EDT-C follows an hybrid approach mixing local and global matchings. To limit the number of triangles and to select the most discriminating ones, EDT-C only uses the triangles resulting from Delaunay triangulations (as in [1], [25], [14], [23]). As reported in [1], [17], with regard to other topologies, Delaunay triangulation has the best structural stability under random positional perturbations.

Unlike existing approaches which assume that each detected minutia point p is genuine, EDT-C not only considers the case where p is genuine, but also the cases where p is spurious, displaced or missing. More precisely, EDT-C uses the triangles resulting from the Delaunay triangulation of the minutiae set and also those resulting from the Delaunay triangulations that would be obtained if each point of the minutiae set was removed. The resulting triangulation is illustrated in figure 1(d) and is called Expanded Delaunay Triangulation [10].

EDT-C describes the built triangles by a set of geometric and fingerprint features sensitive to reflection and robust with regard to non-linear distortions. In addition, EDT-C speeds up the triplets-matching stage through a modified merge-join and the consolidation stage through a voting step.

The remainder of the present paper is organized as follows. Section II briefly reviews prior related work. Section III introduces EDT-C, our algorithm for minutiae-based fingerprint matching. Section IV presents an experimental study of EDT-C on FVC databases. Section V concludes the paper.

II. Background and prior work

A. Minutiae-based matching

In the sequel, we denote a fingerprint acquired by enrollment as *template* T and the query or *input* fingerprint as I. I is said to be *genuine* if it comes from the same finger as T and *impostor* if not [20].

A minutia point is commonly represented by triplet (x, y, θ), where (x, y) represents its spatial location and $\theta \in [0, 2\pi]$ represents its orientation. Each minutia point is typically associated with a tolerance box ($i.e.$ maximum spatial and orientation difference permitted) to compensate spatial changes caused by distortions and extraction errors.

The main idea in minutiae-based matching is to calculate a likeliness score between two fingerprints I and T, according to the number of minutia points from I and T that match. To do so, each minutia point from I is paired with either exactly one minutiae from T, or none at all. This minutiae pairing is commonly preceded by a global alignment of I and T. The alignment consists of rotation, displacement, scaling and other geometric transformations applied to I minutia points.

Finding the correct alignment of I and T is not a trivial task as a minutia point from I may fall within the tolerance box of several minutiae from T [20]. It is widely accepted that the best alignment is the one that maximizes the number of matching minutiae ($i.e.$ minutiae pairs).

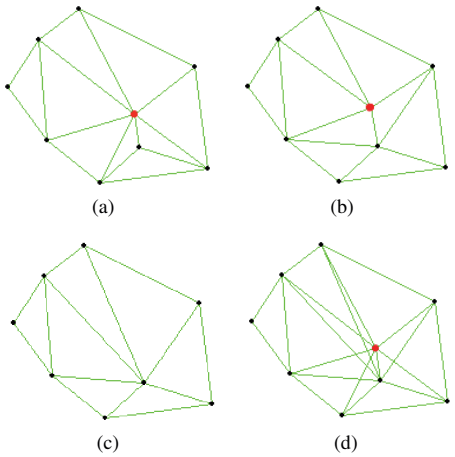


Fig. 1. (a) Delaunay Triangulation. (b) Delaunay Triangulation after the displacement of the red point. (c) Delaunay Triangulation without the red point. (d) Expanded Delaunay Triangulation (EDT). EDT includes the triangles of figures (a) and (c) and most of the triangles of figure (b).

As mentioned before, most of recent minutiae-based fingerprint matching, perform a local minutiae matching followed by a global minutiae matching ($i.e.$ *consolidation*). The purpose of local minutiae matching is to compare local regions ($i.e.$ local structures) from I and T to find out, for each minutia from I, a set of potential matching minutiae from T. The consolidation step tries to derive the transformation(s) that better align I to T based on matching local structures.

B. Triplets-based local structures

Minutiae triplets can be built under different spatial assumptions. [11], [6], [2] consider all possible combinations of triplets in the fingerprint. The use of all possible triplets in a fingerprint containing N minutia points, produces C_N^3 triangles and leads therefore to a high computational cost and to an increased risk of fortuitous (false) matches [5].

[18] uses a Nearest Neighbor (NN) approach. Given a minutia point p, [18] builds all possible triangles formed by p and two of its N nearest neighbors. As reported in [20], NN approaches can be highly affected by missing and spurious minutiae.

In [1], [25], [14] and [23] minutiae triangles are selected from the Delaunay Triangulation (DT) of the entire fingerprint minutiae. DT for a set of points P with cardinality N, is a triangulation $DT(P)$ such that no point in P is inside the circumcircle of each triangle in $DT(P)$ [14]. $DT(P)$ contains $O(N)$ triangles and describes a unique topological structure ($i.e.$ $graph$) of a fingerprint (figure 1(a)). Due to their small number, minutiae triangles can be computed efficiently in $O(NlogN)$ [1]. Furthermore, DT maximize the minimum angle of all angles in the triangulation [1]. These properties are very useful in the context of fingerprint verification. However, as reported in [10], DT is not robust enough with regard to displaced, missing and spurious minutiae, as even a small minutia displacement ($e.g.$ displacement of the red point in figure 1(b)) can severely impacts the DT graph structure.

To minimize the negative effects caused by displaced minutiae, [14] considers a variation of DT named Low-Order

LoD ($i.e.$ order-0 and order-1). An order-r DT of a set of points P, noted $DTr(N)$, is a triangulation such that r points of N are inside the circumcircle of each triangle of $DTr(P)$. [14] proposes to consider all the triangles resulting from the union of $DT_0(P)$ and $DT_1(P)$:

$$LoD(P) = DT_0(P) \cup DT_1(P)$$

While more robust to displaced minutiae than DT, LoD can be highly impacted by missing and spurious minutiae [10]. This illustrated in in figure 1(c) where missing and/or spurious minutiae severely affect the triangulation graph by introducing spurious triangles or eliminating important ones.

In this paper we use a variant of Delaunay Triangulation called Expanded Delaunay Triangulation [10]. Expanded Delaunay Triangulation was first introduced for indexing purpose in [10]. In this paper we show that it can be used as well for fingerprint matching. Expanded Delaunay Triangulation is discussed in detail in section III-A.

C. Consolidation

Consolidation consists in finding the global transformation $(\delta_x, \delta_y, \delta_\theta)$ that best aligns I to T, $i.e.$ that maximizes the number of I and T matching minutiae.

Roughly there exists four consolidation types [17]: single transformation, consensus, incremental and multiple transformations. In single transformation approaches, the transformation aligning the best matching local structures (or a very restricted number of matching local structures) is used to globally align I to T. Single transformation is very efficient as it substantially reduces the search space. However, if the global matching algorithm is not robust enough with regard to displaced and spurious minutiae, the overall robustness will be highly impacted [17].

Consensus consolidation, used in [22], [13], evaluates the consistency of each transformation obtained from a matching pair of local structures with the remaining matching pairs. The main drawback of this approach is that it maximizes the number of matching local structures, rather than the number of matching minutiae [20].

Incremental Transformation [5] arranges local structures of a fingerprint into a graph whose edges represent spatial relationships. The matching is performed by a dual graph traversal. The traversal starts from a given pair, propagates to neighboring nodes, stops when a pair of non-matching nodes is found and returns the number of matching nodes. The process is repeated for each pair of nodes and the maximum number of matched nodes is retained [17]. Incremental transformation is robust but has a high computational cost.

Multiple transformations consolidation [19], [18], [24], [9] consists in testing multiple candidate transformations and: (i) select the one that achieves the highest final score; or (ii) fuse the results obtained by the candidate transformations. Multiple transformations consolidation provides a good trade-off between robustness and efficiency: it is faster than incremental consolidation and more robust than single transformation and consensus consolidations.

III. EDT-C: EDT-BASED MINUTIAE MATCHING

This section introduces EDT-C, a new fingerprint matching algorithm designed to be more tolerant to missing, displaced and spurious minutiae than existing matchers. EDT-C consists of four stages: (i) minutiae triangles generation; (ii) triangles characterization; (iii) triangles pairing; (iii) minutiae pairing; and (iv) consolidation (global matching). The remainder of this section details each of these stages.

A. Local structures determination

EDT-C uses a variant of Delaunay Triangulation, termed *Expanded Delaunay Triangulation* (EDT) [10].

Let P=$\{p_1, p_2, \ldots, p_N\}$ be a set of N minutia points, $DT(P)$ the Delaunay triangulation of P, E the set of $DT(P)$ edges and $G = (P, E)$ the graph described by $DT(P)$.

To be able to define $EDT(P)$, we need first to define the *triangular hulls* [10]. Let p_i be a point of P and $P_i = \{p_j | \{p_i, p_j\} \in E\}$ the set of points of P which are directly connected to p_i in the graph G. The number of points in P_i is the degree of p_i in G. The *triangular hull* H_i of p_i is defined as the Delaunay Triangulation of P_i.

The Expanded Delaunay Triangulation $EDT(P)$ of P is defined as the union of $DT(P)$ triangles and those of the triangular hulls of the points of P.

$$EDT(P) = DT(P) \cup DT(P_1) \cup DT(P_2) \cup \cdots \cup DT(P_N) \quad (1)$$

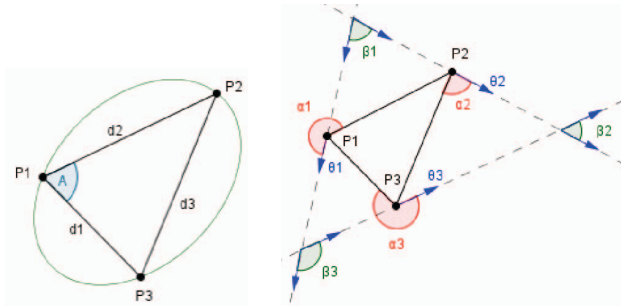
Intuitively, $EDT(P)$ contains, in addition to the triangles of $DT(P)$, all the triangles that would be obtained if each of the points of P was eliminated (figure 1(d)). As $DT(P)$, $EDT(P)$ yields a unique topological structure.

$DT(P)$ based approaches assume that each detected minutia p_i is genuine and do not consider the case where p_i is spurious/missing or displaced. In our approach we consider both cases : $DT(P)$ allows to include all the triangles that would be obtained if p_i was genuine, where $DT(P_i)$ allows to include all the triangles that would be obtained if p_i was spurious/missing or displaced. Accordingly, our approach is expected to be more tolerant to displaced, missing and spurious minutiae than approaches based on Delaunay Triangulation.

The counterpart of EDT robustness is that it constructs more triangles than DT. However, as demonstrated in [10], the number of triangles of $EDT(P)$ remains linear with respect to N and in all cases lower than $|EDT(P)| < 13N - 25$.

B. Triplet descriptor

Each minutiae triangle is represented by a descriptor taking the form of a vector of numerical values (features). Triangles matching and pairing are based on their associated descriptors. The descriptor that we associate with each minutiae triangle follows the recommendations of [18] and includes geometric features that describe the shape of the triangle (figure 2(a)) and features related to the minutiae points that form the triangle (figure 2(b)).



(a) (b)

Fig. 2. (a) Triangle geometric features (elongation of the Steiner circumellipse, cosine of the greatest angle and the perimeter). (b) Fingerprint features [18].

1) Triangle geometric features: Several shape measures can be extracted from a triangle (*e.g.* perimeter, lengths of sides, angles, etc.). With respect to the lengths of sides, angles and relative/derived measures are less sensitive to distortions, but are also less discriminating.

In contrast with most of existing approaches, in our approach we do not describe a minutiae triangle by the lengths of its sides. Instead, we consider a mix of derived and relative measures. To this end, we conducted an experimental study on FVC databases using different combinations of shape measures and retained the one with the best results.

In our approach we describe the shape of a triangle Δ by: the elongation of Δ's Steiner circumellipse, $\epsilon(\Delta)$, the cosine of Δ's greatest angle, $CosA(\Delta)$ and the perimeter of Δ, $sp(\Delta)$. We should notice here that $\epsilon(\Delta)$ and $CosA(\Delta)$ remain invariant if Δ is rotated or scaled uniformly.

Let $d_1 \leq d_2 \leq d_3$ be the lengths of Δ's sides. $\epsilon(\Delta)$, can be calculated using the following formula [8] :

$$\epsilon(\Delta) = \frac{\sigma_1}{\sigma_2}, \text{ where}$$
$$\sigma_1 = \frac{\sqrt{d_1^2+d_2^2+d_3^2+2Z}}{3}, \sigma_2 = \frac{\sqrt{d_1^2+d_2^2+d_3^2-2Z}}{3} \text{ and}$$
$$Z = \sqrt{d_1^4+d_2^4+d_3^4-d_1^2d_2^2-d_1^2d_3^2-d_2^2d_3^2}$$

The cosine of Δ greatest angle can be calculated with the following formula :

$$cosA(\Delta) = min\left(\frac{d_1^2+d_2^2-d_3^2}{2d_1^2d_2^2}, \frac{d_1^2+d_3^2-d_2^2}{2d_1^2d_3^2}, \frac{d_2^2+d_3^2-d_1^2}{2d_2^2d_3^2}\right)$$

2) Minutiae features: Triangle measures taken alone are not sufficient to uniquely identify a minutiae triplet, as they only consider minutiae spatial locations and not their orientations θ_i [18]. As in [18] we consider the following additional minutiae features.

- $\alpha_{i\in[1,3]}$: the angles required to rotate the direction θ_i of a minutia p_i to superpose it to the vector associated with the minutiae (p_i, p_{i+1}).
- $\beta_{i\in[1,3]}$: the angle required to rotate the direction θ_i of a minutia p_i to superpose it to the direction θ_{i+1} of the minutia p_{i+1}.

C. Local matching

As in most existing triangles-based matching approaches, we perform the local matching in two steps : triangles matching and minutiae pairing.

1) Triangles matching: Let I and T be two fingerprints to be compared. The aim of the triangles matching stage is to find the list A of pairs of triangles from I and T that are similar, with respect to a similarity function SL and a threshold Th_{SL} :

$$A = \{(\Delta^I, \Delta^T) | \Delta^I \in I,\ \Delta^T \in T \text{ and } SL(\Delta^I, \Delta^T) \geq Th_{SL}\} \quad (2)$$

Let $S_\epsilon(\Delta^I, \Delta^T)$, $S_{CosA}(\Delta^I, \Delta^T)$, $S_{sp}(\Delta^I, \Delta^T)$, $S_\alpha(\Delta^I, \Delta^T)$ and $S_\beta(\Delta^I, \Delta^T)$ be, respectively, the functions allowing to calculate the similarity between elongations, cosines of the largest angles, the perimeters, the angles α and β of two triangles Δ^I and Δ^T.

$S_\epsilon(\Delta^I, \Delta^T)$, $S_{CosA}(\Delta^I, \Delta^T)$ and $S_{sp}(\Delta^I, \Delta^T)$ are respectively calculated according to the following self-explanatory formulas.

$$S_\epsilon(\Delta^I, \Delta^T) = \begin{cases} 0 \ \ if \ \ |\epsilon(\Delta^I) - \epsilon(\Delta^T)| \ > \ Thr_\epsilon \\ 1 - \frac{|\epsilon(\Delta^I) - \epsilon(\Delta^T)|}{Thr_\epsilon}, \quad otherwise \end{cases} \quad (3)$$

$$S_{cosA}(\Delta^I, \Delta^T) = \begin{cases} 0 \ if \ |cosA(\Delta^I) - cosA(\Delta^T)| > Thr_{cosA} \\ 1 - \frac{|cosA(\Delta^I) - cosA(\Delta^T)|}{Thr_{cosA}}, \quad otherwise \end{cases} \quad (4)$$

$$S_{sp}(\Delta^I, \Delta^T) = \begin{cases} 0 \ \ if \ \ |sp(\Delta^I) - sp(\Delta^T)| \ > \ Thr_{sp} \\ 1 - \frac{|sp(\Delta^I) - sp(\Delta^T)|}{Thr_{sp}}, \quad otherwise \end{cases} \quad (5)$$

$$S_\alpha(\Delta^I, \Delta^T) = \begin{cases} 0 \ if \ \exists i \in [1,3] \mid \left(ad(\alpha_i^I, \alpha_i^T) \ > \ Thr_a\right) \\ 1 - \frac{max_{i\in[1,3]}\{ad(\alpha_i^I, \alpha_i^T)\}}{Thr_a}, \quad otherwise \end{cases} \quad (6)$$

$$S_\beta(\Delta^I, \Delta^T) = \begin{cases} 0 \ if \ \exists i \in [1,3] \mid \left(ad(\beta_i^I, \beta_i^T) \ > \ Thr_a\right) \\ 1 - \frac{max_{i\in[1,3]}\{ad(\beta_i^I, \beta_i^T)\}}{Thr_a}, \quad otherwise \end{cases} \quad (7)$$

The tolerance boxes defined by the thresholds (Thr_ϵ, Thr_{cos}, Thr_{sp} and Thr_a) are necessary to compensate the unavoidable errors made during fingerprint acquisition. As in [18], for two given angles a_1 and a_2, we use the function $ad(a_1, a_2) = min\left(|a_1 - a_2|, 2\pi - |a_1 - a_2|\right)$ to compute the minimum angle required to superpose two vectors with the same origin and angles a_1 and a_2, respectively.

The similarity between two triangles $\Delta^I \in I$ and $\Delta^T \in T$ is calculated by the following formula.

$$SL(\Delta^I, \Delta^T) =$$

$$\begin{cases} 0 \; if \; S_\epsilon(\Delta^I, \Delta^T) = 0 \vee S_{cosA}(\Delta^I, \Delta^T) = 0 \\ \qquad \vee S_{sp}(\Delta^I, \Delta^T) = 0 \vee S_\alpha(\Delta^I, \Delta^T) = 0 \\ \qquad \vee S_\beta(\Delta^I, \Delta^T) = 0 \\ 1 - \left(1 - S_\epsilon(\Delta^I, \Delta^T)\right)\left(1 - S_{cosA}(\Delta^I, \Delta^T)\right) \\ \qquad \left(1 - S_{sp}(\Delta^I, \Delta^T)\right)\left(1 - S_\alpha(\Delta^I, \Delta^T)\right) \\ \qquad \left(1 - S_\beta(\Delta^I, \Delta^T)\right) \qquad otherwise \end{cases} \tag{8}$$

If $SL(\Delta^I, \Delta^T) \leq Th_{SL}$, the pair (Δ^I, Δ^T) is added to the list A. Otherwise, the pair is rejected. To speed up local structures matching, the triangles of I and those of T are maintained sorted according to their perimeters and a slightly modified merge-join is used to determine matching triangles.

2) Local minutiae pairing: The goal of this stage is to build a list M containing the pairs of minutiae from I and T that match : $M = \{((p,q),v) \,|p \in I,\, q \in T \; and \; v > 0\}$, where v corresponds to the number of triangles where p and q match. In other words, M is used to accumulate votes or evidences on minutiae matching.

M is constructed as follows :

1. For each pair of matching triangles $(\Delta^I, \Delta^T) \in A$, we consider the minutiae triplets forming them : $(p_1, p_2, p_3) \in \Delta^I$ and $(q_1, q_2, q_3) \in \Delta^T$.
2. For each minutiae mates (p_i, q_j), if p_i and q_j match and (p_i, q_j) appears in M, then the number of occurrences ($i.e.$ votes) associated to (p_i, q_j) is incremented by one.
3. If p_i and q_j match and (p_i, q_j) does not appear in M, $((p_i, q_j), 1)$ is added to M

D. Consolidation

Consolidation is the ultimate stage in our approach. EDT-C uses a slightly modified multiple transformations consolidation, aiming at achieving a good trade off between robustness and efficiency. In our approach we consider the m minutiae pairs of M that have the highest v ($i.e.$ highest numbers of votes). Each transformation that align a minutiae pair from m, is considered as a candidate transformation. Rather than testing each candidate transformation on the entire set of minutiae, we only tested it on the m pairs with the highest votes. The transformation that better align the m pairs, is then applied at the global level and the similarity score is calculated accordingly.

Let $|I|$ and $|T|$ be respectively the number of minutia points in the enrolled and the query fingerprints, and k the number of matching minutiae. The similarity score is calculated using the formula [17][20].

$$Score(I, T) = \frac{k}{(|I| + |T|)/2} \tag{9}$$

IV. Experimental results

FVC databases [15], [16], [3] are commonly used as benchmarks for evaluating fingerprint verification algorithms. To validate our work, we used FVC protocol and databases to compute the following accuracy indicators:

Triangulation	EER%	FMR100%	FMR1000%	zeroFMR%	Temps(ms)
NN	2,500	3,179	4,893	6,750	4,059
DT	2,358	3,143	4,679	5,321	**1,195**
LoD	2,256	2,821	4,107	6,500	1,703
EDT	**1,572**	**1,893**	**2,893**	**3,357**	4,378

TABLE I. Experimental results on database DB1_A of FVC2002.

Algorithm	Triplets	Shape measures	Consolidation
PN	Delaunay Triangulation	Lengths of sides	Multiple transformations
M3gl	N nearest neighboring minutiae	Lengths of sides	Multiple transformations
EDT-C	Expanded Delaunay Triangulation	Elongation of the $Steiner$ circumellipse, Cosine of the greatest angle and the perimeter	Multiple transformations (only on the m matching pairs with the highest votes)

TABLE II. Main differences between PN[19], M3gl[18] and EDT-C

- FMR: *False Match Rate*, rate of incorrectly matched fingerprints. Each score threshold has an associated FMR.
- FNMR: *False Non-Match Rate*, rate of corresponding fingerprints that are incorrectly considered different. Each score threshold has an associated FNMR.
- EER: *Equal-Error Rate*, score threshold where FMR and FNMR are equal.
- FMR100: lowest achievable FNMR for an FMR $\leq$ 1%.
- FMR1000: lowest achievable FNMR for an FMR $\leq$ 0.1%.
- ZeroFMR: lowest FNMR at which no false matches occur.
- ROC: curve that plots the FNMR as function of the FMR.

Experiments were performed on a dedicated dual core i3-2330M system. Each core offers a base speed of 2.2 GHz. The computer features 4 GB main memory (DDR3-1066/1333) and 3 MB L3 cache. The indicator Time is used in the sequel to refer to the average matching time in milliseconds.

DB	Algorithm	EER%	FMR100%	FMR1000%	ZeroFMR%	Time(ms)
DB1_A	PN	3,657	4,679	7,071	12,750	50,662
	M3gl	2,406	3,071	5,286	6,964	**3,296**
	EDT-C	**1,572**	**1,893**	**2,893**	**3,357**	4,378
DB2_A	PN	2,272	2,750	3,750	4,286	61,088
	M3gl	1,716	1,893	3,036	4,643	**3,400**
	EDT-C	**1,122**	**1,250**	**1,964**	**2,500**	4,341
DB3_A	PN	5,944	8,357	12,500	16,107	182,105
	M3gl	5,726	8,929	12,536	13,893	**6,201**
	EDT-C	**4,228**	**6,036**	**7,750**	**13,464**	8,656
DB4_A	PN	5,051	5,857	7,250	7,964	15,730
	M3gl	2,498	3,286	5,571	**8,179**	1,706
	EDT-C	**2,074**	**2,714**	**4,893**	9,250	**1,403**

TABLE III. Experimental results on FVC2000 databases

DB	Algorithm	EER%	FMR100%	FMR1000%	ZeroFMR%	Time(ms)
	PN	1,355	1,393	2,536	4,214	39,885
DB1_A	M3gl	0,995	**1,000**	1,714	3,643	**2,833**
	EDT-C	**0,955**	**1,000**	**1,464**	**1,964**	4,782
	PN	1,622	1,821	2,857	3,929	107,707
DB2_A	M3gl	0,902	1,036	1,500	1,786	**4,245**
	EDT-C	**0,632**	**0,679**	**0,821**	**1,143**	8,514
	PN	5,210	7,250	9,929	15,143	14,731
DB3_A	M3gl	4,142	5,464	9,429	**11,286**	**1,335**
	EDT-C	**3,569**	**5,000**	**7,250**	13,357	1,662
	PN	2,426	3,179	6,500	9,179	22,753
DB4_A	M3gl	2,046	2,643	5,714	**8,036**	**2,654**
	EDT-C	**1,305**	**1,429**	**4,000**	8,357	3,026

TABLE IV. EXPERIMENTAL RESULTS ON FVC2002 DATABASES

DB	Algorithm	EER%	FMR100%	FMR1000%	ZeroFMR%	Time(ms)
	PN	7,640	11,179	15,214	19,321	44,006
DB1_A	M3gl	**4,434**	**7,893**	14,893	18,357	**3,264**
	EDT-C	5,125	8,393	**12,964**	**14,500**	5,846
	PN	8,026	10,036	12,607	15,500	35,590
DB2_A	M3gl	5,838	8,321	11,429	13,607	**3,444**
	EDT-C	**5,059**	**6,536**	**9,571**	**10,071**	4,269
	PN	5,825	7,750	15,714	18,286	89,485
DB3_A	M3gl	4,150	6,536	10,357	12,536	**5,261**
	EDT-C	**3,645**	**4,786**	**7,107**	**12,500**	8,024
	PN	4,094	5,143	7,071	10,643	45,537
DB4_A	M3gl	2,566	3,607	5,786	6,571	**2,865**
	EDT-C	**2,148**	**2,964**	**5,000**	**5,857**	4,184

TABLE V. EXPERIMENTAL RESULTS ON FVC2004 DATABASES

We conducted two sets of experiments. The aim of the first one is to bring to light the relevancy of using Expanded Delaunay Triangulation (EDT) for fingerprint verification purpose (without any other consideration). The aim of the second set is to compare EDT-C to its main competitors. To show the relevancy of using Expanded Delaunay Triangulation, we implemented our approach with alternative triplets generation methods, namely : the N nearest neighbors (NN), the Delaunay Triangulation (DT) and the combined Low-order Delaunay Triangulations (LoD). The results obtained on the database DB1_A of FVC 2002 are reported in table I. As expected, although EDT consumes a slightly higher execution time, it is by far more accurate than any other topology.

DB	Algorithm	EER%	FMR100%	FMR1000%	ZeroFMR%	Time(ms)
	PN	0,975	0,942	1,764	3,452	196,373
DB2_A	M3gl	0,933	0,920	1,623	2,197	**4,872**
	EDT-C	**0,448**	**0,346**	**0,671**	**1,028**	8,786
	PN	6,161	7,695	10,584	16,494	98,584
DB3_A	M3gl	4,898	7,998	16,786	29,329	**3,742**
	EDT-C	**4,36**	**5,812**	**8,647**	**14,751**	5,507
	PN	4,017	5,054	9,058	12,619	56,258
DB4_A	M3gl	3,686	5,141	8,885	12,403	**2,609**
	EDT-C	**2,436**	**2,998**	**5,108**	**8,626**	3,576

TABLE VI. EXPERIMENTAL RESULTS ON FVC2006 DATABASES

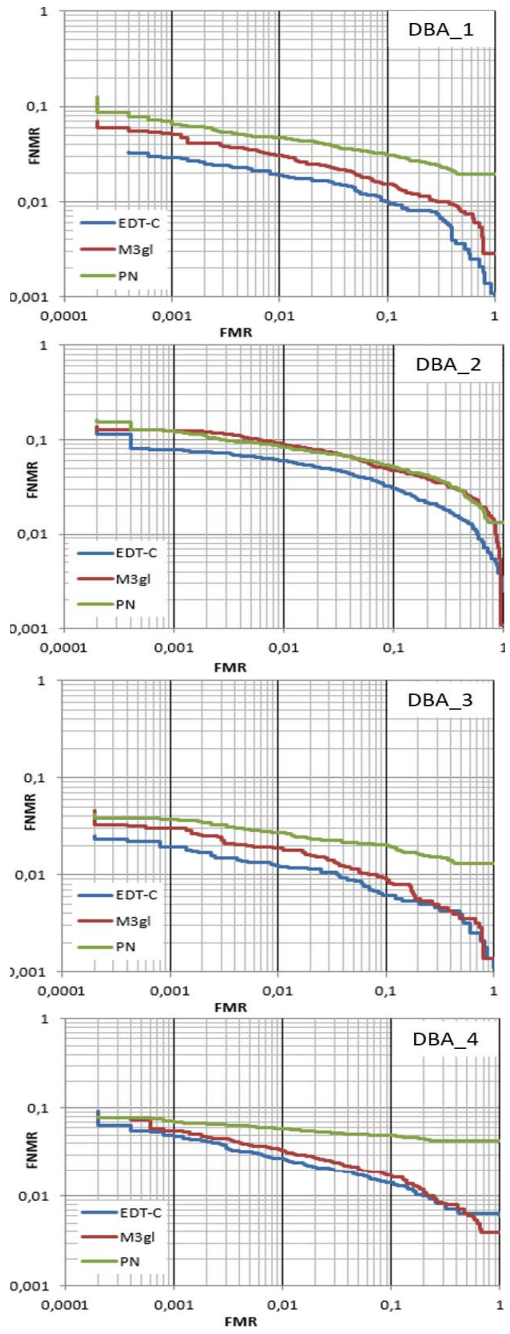


Fig. 3. ROC curves obtained in FVC2000 databases

A large number of experiments have been performed to compare EDT-C to its competitors. Due to the lack of space, only few results are presented herein. In our experimental comparison we include the algorithm proposed by Parziale and Niel [19] (PN) and the algorithm proposed by Medina-Prez *et al*. [18] (M3gl). We implemented PN because it is one of the most popular algorithm based on minutiae triplets in the literature. We implemented M3gl because it proves itself as one of the best triplets-based matchers [18].

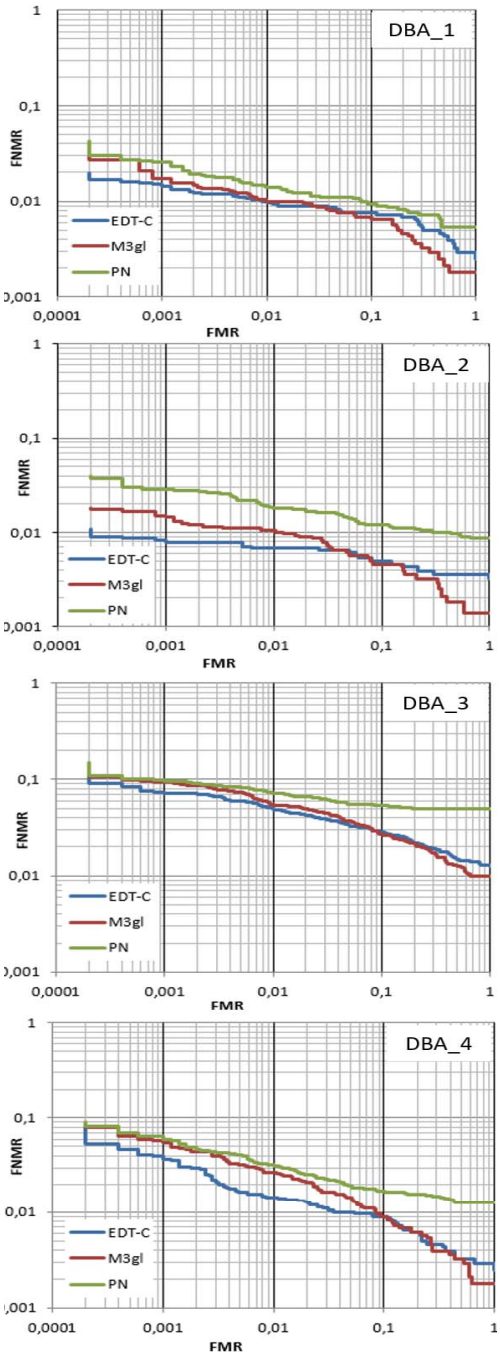


Fig. 4. ROC curves obtained in FVC2002 databases

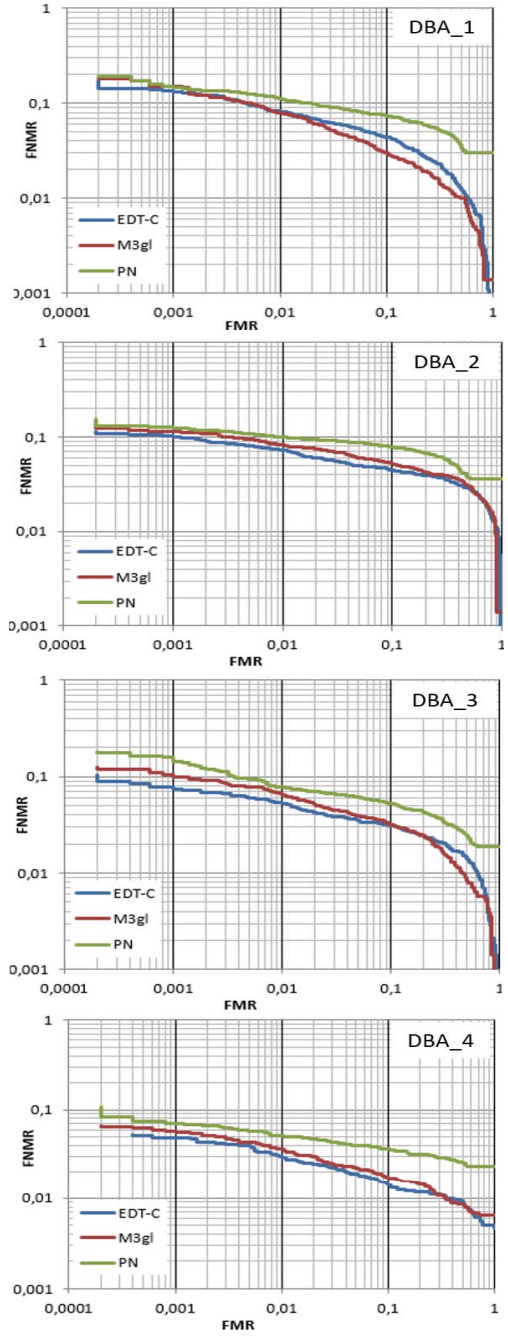


Fig. 5. ROC curves obtained in FVC2004 databases

EDT-C is very close to PN and M3gl in design philosophy. As EDT-C, PN and M3gl: (i) only use standard minutiae features ($i.e.$ spatial location, orientation and type) and, hence, are suitable for systems based on interoperability standards; (ii) use minutiae triangles as local structures; and (iii) perform a local matching and consolidate it by a global matching. PN and M3gl use respectively DT and NN approaches to construct minutiae triangles. Table II summarizes the main properties of the three approaches. A salient feature of M3gl is that it arranges minutiae in clockwise sense and performs the three possible rotations of the triplets to ensure invariance to the order of minutiae and sensitivity to the reflection of minutiae triplets [18]. As illustrated in tables III, IV, V and VI, EDT-C and M3gl outperform PN in almost all databases. This can be explained by the fact that PN uses Delaunay Triangulation and is hence highly impacted by displaced, missing and spurious minutiae. M3gl is slightly faster than EDT-C. However, EDT-C is more accurate in almost all databases. This result is confirmed in figures IV and IV and IV which show that EDT-C achieves lower FNMR for most of the FMR values.

The good performances of EDT-C and its slight additional computational cost can be explained by the fact that Expanded

Delaunay Triangulation generates more triangles than DT and NN, but helps in better support missing, spurious and displaced minutiae.

V. Conclusion and Future Work

Displaced, missing and spurious minutiae make fingerprint matching a very challenging pattern recognition problem. In this paper we present EDT-C, a new minutiae-based matcher. As most of recent algorithms, EDT-C performs a local minutiae matching followed by a consolidation stage. In contrast with existing approaches, EDT-C uses an Expanded Delaunay Triangulation (EDT) to yield local structures. As shown in this paper, EDT is very robust with regard to missing and spurious minutiae. The robustness of EDT can be explained by the fact that it constructs local structures around a minutiae p by taking into account the case where p is genuine as well as the case where it is not. EDT-C uses also a set of innovative features to characterize minutiae triplets. The considered features are more tolerant to displaced minutiae than the usually used lengths of sides. Finally, to fast the entire matching process, EDT-C uses a modified merge-join to speed up local structures matching and a lightweight multiple transformations consolidation.

Experiments conducted on FVC databases show that, while EDT-C has comparable execution time with other triplets-based algorithms, it is by far more accurate. Encouraged by the good behavior of EDT-C, our next goal is to adapt it to latent fingerprint identification.

Acknowledgements : This research paper was performed in the framework of a MOBIDOC thesis funded by the European Union under the program PASRI managed by the ANPR.

References

[1] George Bebis, Taisa Deaconu, and Michael Georgiopoulos. Fingerprint identification using delaunay triangulation. In *Information Intelligence and Systems, 1999. Proceedings. 1999 International Conference on*, pages 452–459. IEEE, 1999.

[2] Soma Biswas, Nalini K Ratha, Gaurav Aggarwal, and Jonathan Connell. Exploring ridge curvature for fingerprint indexing. In *Biometrics: Theory, Applications and Systems, 2008. BTAS 2008. 2nd IEEE International Conference on*, pages 1–6. IEEE, 2008.

[3] Raffaele Cappelli, Matteo Ferrara, Annalisa Franco, and Davide Maltoni. Fingerprint verification competition 2006. *Biometric Technology Today*, 15(7):7–9, 2007.

[4] Raffaele Cappelli, Matteo Ferrara, and Davide Maltoni. Minutia cylinder-code: A new representation and matching technique for fingerprint recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12):2128–2141, 2010.

[5] Sharat Chikkerur and Nalini Ratha. Impact of singular point detection on fingerprint matching performance. In *Fourth IEEE Workshop on Automatic Identification Advanced Technologies (AutoID'05)*, pages 207–212. IEEE, 2005.

[6] Kyoungtaek Choi, Dongjae Lee, Sanghoon Lee, and Jaihie Kim. An improved fingerprint indexing algorithm based on the triplet approach. In *International Conference on Audio-and Video-Based Biometric Person Authentication*, pages 584–591. Springer, 2003.

[7] Sarat C Dass. Fingerprint-based recognition. *International Statistical Review*, 81(2):175–187, 2013.

[8] Gerald Farin. Shape measures for triangles. *IEEE transactions on visualization and computer graphics*, 18(1):43–46, 2012.

[9] Jianjiang Feng. Combining minutiae descriptors for fingerprint matching. *Pattern Recognition*, 41(1):342–352, 2008.

[10] AndréS Gago-Alonso, José HernáNdez-Palancar, Ernesto RodríGuez-Reina, and Alfredo MuñOz-BriseñO. Indexing and retrieving in fingerprint databases under structural distortions. *Expert Systems with Applications*, 40(8):2858–2871, 2013.

[11] Robert S Germain, Andrea Califano, Scott Colville, et al. Fingerprint matching using transformation parameter clustering. *IEEE Computational Science and Engineering*, 4(4):42–49, 1997.

[12] Mohamed Hedi Ghaddab, Khaled Jouini, and Ouajdi Korbaa. Fusion de minuties pour une reconnaissance efficiente des empreintes digitales. In *Colloque sur l'Optimisation et les Systmes d'Information*, 2017.

[13] Xiaoguang He, Jie Tian, Liang Li, Yuliang He, and Xin Yang. Modeling and analysis of local comprehensive minutia relation for fingerprint matching. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(5):1204–1211, 2007.

[14] Xuefeng Liang, Arijit Bishnu, and Tetsuo Asano. A robust fingerprint indexing scheme using minutia neighborhood structure and low-order delaunay triangles. *IEEE Transactions on Information Forensics and Security*, 2(4):721–733, 2007.

[15] Dario Maio, Davide Maltoni, Raffaele Cappelli, James L Wayman, and Anil K Jain. Fvc2002: Second fingerprint verification competition. In *Pattern recognition, 2002. Proceedings. 16th international conference on*, volume 3, pages 811–814. IEEE, 2002.

[16] Dario Maio, Davide Maltoni, Raffaele Cappelli, Jim L Wayman, and Anil K Jain. Fvc2004: third fingerprint verification competition. In *Biometric Authentication*, pages 1–7. Springer, 2004.

[17] Davide Maltoni, Dario Maio, Anil Jain, and Salil Prabhakar. *Handbook of fingerprint recognition*. Springer Science & Business Media, 2009.

[18] Miguel Angel Medina-Pérez, Milton García-Borroto, Andres Eduardo Gutierrez-Rodríguez, and Leopoldo Altamirano-Robles. Improving fingerprint verification using minutiae triplets. *Sensors*, 12(3):3418–3437, 2012.

[19] Giuseppe Parziale and Albert Niel. A fingerprint matching using minutiae triangulation. In *Biometric Authentication*, pages 241–248. Springer, 2004.

[20] Daniel Peralta, Mikel Galar, Isaac Triguero, Daniel Paternain, Salvador García, Edurne Barrenechea, José M Benítez, Humberto Bustince, and Francisco Herrera. A survey on fingerprint minutiae-based local matching for verification and identification: Taxonomy and experimental evaluation. *Information Sciences*, 315:67–87, 2015.

[21] Jin Qi and Yangsheng Wang. A robust fingerprint matching method. *Pattern Recognition*, 38(10):1665–1671, 2005.

[22] Nalini K Ratha, Ruud M Bolle, Vinayaka D Pandit, and Vaibhav Vaish. Robust fingerprint authentication using local structural similarity. In *Applications of Computer Vision, 2000, Fifth IEEE Workshop on.*, pages 29–34. IEEE, 2000.

[23] Arun Ross and Rajiv Mukherjee. Augmenting ridge curves with minutiae triplets for fingerprint indexing. In *Defense and Security Symposium*, pages 65390C–65390C. International Society for Optics and Photonics, 2007.

[24] Lifeng Sha, Feng Zhao, and Xiaoou Tang. A two-stage fusion scheme using multiple fingerprint impressions. In *Proceedings of the International Conference on Image Processing, San Antonio, Texas, USA*, pages 385–388, 2007.

[25] Tamer Uz, George Bebis, Ali Erol, and Salil Prabhakar. Minutiae-based template synthesis and matching using hierarchical delaunay triangulations. In *Biometrics: Theory, Applications, and Systems, 2007. BTAS 2007. First IEEE International Conference on*, pages 1–8. IEEE, 2007.

[26] Akhil Vij and Anoop Namboodiri. Learning minutiae neighborhoods: A new binary representation for matching fingerprints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 64–69, 2014.

.

Autres publications

.

1 Real-Time, CNN-Based Assistive Device for Visually Impaired People

Khaled Jouini, Mohamed Hédi Maaloul & Ouajdi Korbaa
International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI). IEEE. 2021.
DOI : 10.1109/CISP-BMEI53629.2021.9624387

Lien : https://ieeexplore.ieee.org/document/9624387

Real-Time, CNN-Based Assistive Device for Visually Impaired People

Khaled Jouini
Mohamed Hédi Maaloul
and Ouajdi Korbaa

MARS Research Lab LR17ES05
ISITCom
University of Sousse, Tunisia.

***Abstract*—Visual impairment limits people's ability to move about unaided and interact with the surrounding world. This paper aims to leverage recent advances in deep learning to assist visually impaired people in their daily challenges. The high accuracy of deep learning comes at the expense of high computational requirements for both the training and the inference phases. To meet the computational requirements of deep learning, a common approach is to move data from the assistive device to distant servers ($i.e.$ cloud-based inference). Such data movement requires a fast and active network connection and raises latency, cost, and privacy issues. In contrast with most of exiting assistive devices, in our work we move the computation to where data resides and opt for an approach where inference is performed directly "on" device ($i.e.$ on-device-based inference). Running state-of-the-art deep learning models for a real-time inference on devices with limited resources is a challenging problem that cannot be solved without trading accuracy for speed (no free lunch). In this paper we conduct an extensive experimental study of 12 state-of-the-art object detectors, to strike the best trade-off between speed and accuracy. Our experimental study shows that by choosing the right models, frameworks, and compression techniques, we can achieve decent inference speed with very low accuracy drop.**

I. Introduction

Visual impairment limits people's ability to interact with the surrounding world and can negatively affect their socio-economic integration. The World Health Organization estimates that there are approximately 285 million visually impaired and blind worldwide [1]. The prevalence of vision impairment in low- and middle-income regions is estimated to be four times higher than in high-income regions [1]. A common goal in computer vision research is to mimic human visual system and automate tasks that it can perform ($e.g.$, scene recognition, object detection, etc.). In this work, we are interested in leveraging computer vision and deep learning methods to implement an inexpensive yet effective device to assist visually impaired people in their daily challenges.

Object recognition is the cornerstone of devices intended to aid visually impaired people. It consists in drawing a bounding box around each object in an image and associating with each box the class of the surrounded object and a confidence score. Modern object detectors, such as SSD ($i.e.$ Single Shot multi-box Detector) models [2] and YOLO ($i.e.$ You Only Look Once) models [3], [4], [5], [6] are typically built on deep CNNs (Convolutional Neural Networks). In contrast with traditional computer vision methods, modern detectors perform the tasks of feature extraction, object localization, and object classification in one go ($i.e.$, End-To-End).

Due to recent advances that partially solved the vanishing gradient problem ($e.g.$, residual blocs [7]), current CNNs are becoming deeper and more complex (up to tens of layers). Such CNNs can match or even exceed human vision in accuracy and have, by far, outperformed traditional computer vision methods in a variety of fields and in various challenges [8]. However, the high accuracy of deep CNNs comes at the expense of high computational and memory requirements for both the training and the inference phases. Training is computationally expensive due to millions of parameters that need to be iteratively refined [9]. Inference is expensive due to billions of multiply-accumulate (MACs) operations that need to be performed on input data [9]. Due to their high computational requirements, it is difficult to execute deep CNNs on embedded and mobile devices with limited resources [9], [10].

There exist several CNN-based systems designed to support visually impaired people [11], [12], [13], [14]. Unfortunately, most of them inherit the drawbacks of deep CNNs and fail to provide a full on-device, real-time object recognition.

In this work, we intend to provide visually impaired people with an actual real-time on-device inference. In particular, this paper focuses on reviewing and studying recent lightweight CNN-based object detectors. Lightweight CNN-based object detectors aim to be faster and less demanding on resources than conventional CNNs, at the expense of a "small" accuracy loss. Used in conjunction with compression techniques, lightweight architectures make feasible the deployment of deep CNNs on devices with constrained resources. The purpose of our study is to strike the best trade-off between speed and accuracy. To the best of our knowledge, there exists no previous work on assistive devices that compares or uses lightweight CNN architectures. Our work's main contributions are therefore: (i) the implementation of an inexpensive device, specifically tailored for visually impaired people and allowing an effective real-time inference; and (ii) a study on the running of lightweight object detectors on end devices.

The remainder of this paper is organized as follows. Section II briefly reviews previous work on CNN-based object detectors and CNN-based assistive devices. Section III details the implementation steps of our system and presents an extensive experimental study of state-of-the-art lightweight detectors. Finally, section IV concludes the paper.

II. Background

A. One-stage object detectors

Early CNN-based object detectors like R-CNN [15], perform the task of object detection in two stages. They first generate region proposals ($i.e.$, areas that potentially contain an object) and then run a classifier on the proposed regions [15]. While the accuracy of two-stages detectors is often good, they are relatively slow since they require running the model's detection and classification portions several times. Recent one-stage detectors, in contrast, perform only a single pass through the CNN and simultaneously predict multiple bounding boxes and class probabilities in one go. One-stage detectors are in general, faster than two-stage detectors while achieving comparable accuracy.

One-stage detectors are based on two ideas. The first idea consists in framing detection as a regression problem. The second is to use a fixed-size grid of detectors. Two of the most popular one-stage detectors are YOLO [3] and SSD [2].

1) YOLO (You only Look Once): The YOLO method, introduced by Redmon et al. [3] in 2016, was the first proposed one-stage object detector. In its original implementation, YOLO divides spatially each input image into a fixed grid of $S \times S$ cells. Each grid cell predicts k bounding boxes and confidence scores for those boxes. YOLO therefore always predicts the same number of bounding boxes ($S \times S \times k$). As post-processing, boxes having confidence scores below a predefined threshold are dropped out. The remaining boxes undergo a non-maximum suppression to eliminate eventual duplicate detections.

Redmon et al. also proposed YOLOv2 [4] and YOLOv3 [5], respectively in 2017 and 2018. YOLOv2 and YOLOv3 implement several techniques to improve speed and/or accuracy ($e.g.$, batch normalization, k-means, etc.). In particular, YOLOv2 predefines k different box shapes, called anchor boxes, that correspond to the k most common object shapes in the dataset. YOLOv2 chooses anchors by running k-means clustering on all the bounding boxes from all the training images. Each object in a training image is then assigned to the grid cell that contains the object's midpoint and to the anchor box with the highest overlap divided by non-overlap (called Intersect-over-Union, or IoU). While the fixed grid forces the model to learn specialized object detectors in specific locations, anchors force the detectors inside the cells to each specialize in a particular object shape. A salient feature of YOLOv3, the last YOLO version proposed by Redmon et al., is that it uses three different scale grids ($i.e.$ three different scale feature maps) to better handle objects of various scales.

Most of existing CNN-based object detectors are composed of at least two parts, a backbone for feature extraction and a head for object classification and bounding box regression. Some of the most recent object detectors insert in addition some layers, called neck, between the backbone and the head. The neck part is used to collect feature maps from different stages [6]. YOLOv2 uses a custom CNN named Darknet-19 (having 19 layers) as backbone. YOLOv3 is based on a deeper version of Darknet named Darknet-53 (having 53 layers). In addition, YOLOv3 uses FPN (Feature Pyramid Network) as neck to extract features of different scales from the backbone.

TABLE I
Number of layers in regular and lightweight YOLO versions.

Model	# of Layers	Pretrained Convolutional Weights
YOLOv2	32	199 MB
YOLOv2-Tiny	16	43 MB
YOLOv3	106	236 MB
YOLOv3-Tiny	24	34 MB
YOLOv4	162	245 MB
YOLOv4-Tiny	38	23 MB

Bochkovskiy et al. proposed YOLOv4 [6] in 2020. YOLO v4 is an evolution of YOLOv3 that achieves state-of-the-art results by combining two categories of methods: Bag of Freebies (BoF) and Bag of Specials (BoS). BoF refers to methods, such as Mosaic and CutMix data augmentation, DropBlock regularization, and CIoU loss, that improve accuracy without increasing the inference time. BoS refers to methods, such as Weighted-Residual-Connections (WRC), Cross-Stage-Partial-connections (CSP), and Mish-activation, that slightly increase the inference time but significantly improve accuracy. YOLOv4 uses CSPDarknet53 [16] as backbone, Spatial pyramid pooling (SPP), and Path Aggregation Network (PAN) as neck, and the same head as YOLOv3. YOLOv4 also selects optimal hyper-parameters by applying genetic algorithms.

Along with regular versions, lightweight versions of YOLO (with the suffix "-tiny") were also proposed. Lightweight YOLO versions use less convolutional and pooling layers than regular architectures (Table I), which results in a higher processing speed, but also in a reduced accuracy. YOLOv3-tiny and YOLOv4-tiny, in addition, use two feature maps instead of the three used in the regular corresponding models.

2) SSD (Single Shot MultiBox Detector): Conceptually, YOLO, and SSD [2] follow the same principles and differ only in details. To detect objects of various scales, the input image in SSD is passed through a series of convolutional layers, generating several sets of feature maps at different scales. SSD also uses different default bounding boxes (anchors) with different scales and shapes (aspect ratios) and then adjusts the bounding boxes as part of the prediction. In contrast with YOLO, SSD's default bounding boxes ($i.e.$ anchors) are independent of the dataset. In different feature maps, the scale of default anchors is computed with regularly space between the highest layer and the lowest layer.

Like YOLO, SSD always produces the same number of bounding boxes and performs a non-maximum suppression to yield the final predictions. In contrast with YOLO, SSD is designed to be independent of the underlying backbone. In the original SSD implementation, the VGG-16 [2] model was

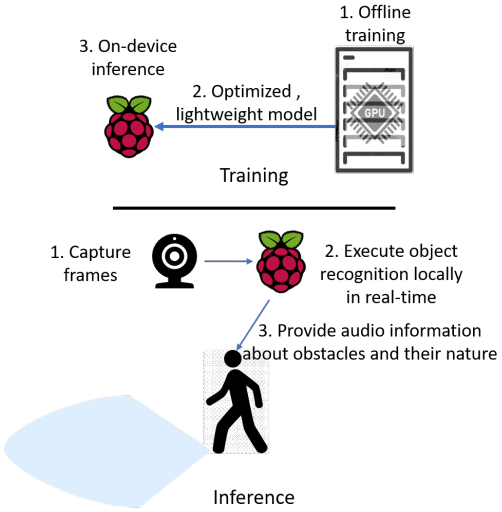


Fig. 1. Inference and training are decoupled.

Fig. 2. The proposed system takes the form of a camera harness incorporating a nanocomputer.

used as backbone. In this research, we use MobileNet [2], instead. MobileNet uses depth-wise separable convolutions to build lightweight networks, suitable for mobile and embedded systems. The association of MoblileNet and SSD for (a lightweight) object detection was suggested in [2] and called SSD Lite. Its main strength is that it uses depth wise separable convolutions for the SSD layers as well to speed up the head part of the network.

B. Evaluation of the accuracy of object detectors

Evaluating the accuracy of object detectors is non trivial because there are two distinct tasks to measure: (i) a classification task ($i.e.$ determining the nature of an object); and (ii) a regression task ($i.e.$ determining the bounding box of an object). The regression task is typically evaluated by the IoU, which measures the overlap between a predicted bounding box and the corresponding ground truth bounding box. A detection is considered to be a true positive only if the predicted class matches the actual class, and the predicted bounding box has an IoU greater than a predefined threshold with the ground truth bounding box. If one of the above two conditions is not satisfied, the detection is considered to be a false positive.

The most commonly used accuracy evaluation metric in object detection is the "Average Precision (AP)" [1], whose values range from 0 to 100 (higher values are better). For the COCO challenge [17], the AP score is averaged over 80 classes and 10 IoU thresholds, ranging from 0.5 to 0.95 with a step size of 0.05. The COCO AP is written as AP@[.50:.05:.95] or simply as AP, while the AP at fixed IoU such as IoU=0.5, is written as AP@0.5. Averaging over 10 IoU thresholds instead of considering a single IoU threshold of .50 (as in PASCAL VOC challenge) tends to reward models that are better at precise localization.

C. CNN-Based assistive devices for visually impaired people

Existing CNN-based assistive systems providing on-device inference, such as [11], often have very limited capabilities ($e.g.$, recognition limited to faces, stairs, etc.). On the other hand, existing systems providing full object recognition capabilities either use GPU-based computers, unsuitable for convenient mobility [12], [13], or use a mobile/embedded device, but rely on remote servers (cloud-based inference). Moving data for inference from the source to the cloud requires an active and fast internet connection, and raises latency, cost, privacy, and scalability issues [18]. As an example, real experiments showed that offloading a camera frame to a cloud server and executing an image classification task, takes from 350 ms to more than 600 ms end-to-end ($\approx$ 1.67 to $\approx$ 2.86 Frame Per Second) [18].

The most recent work on CNN-based assistive devices [14], opts for an on-device inference and uses YOLOv2-tiny for object recognition and MTCNN [19] for facial identification. The paper, however, does not investigate Speed/Accuracy trade-offs to identify the best-suited lightweight object detector. As shown in section III, YOLOv2-tiny is the least accurate tested model (figure 3), while not being the fastest one.

III. System implementation and performance benchmarking

A. System overview

The architecture of our solution is depicted in 1. We implemented our system on a Raspberry Pi 4B having a 1.5 GHz 64-bit SoC quad core ARM Cortex-A72 (ARM v8) processor, 8 GB LPDDR4-3200 SDRAM and a 8 MP Raspberry Pi V2 camera. The system portability is achieved using a rechargeable battery. We found that this $\approx$ 80$ nanocomputer offers a good trade-off between cost, practicality, and computation power.

We implemented a Text-to-Speech (TTS) engine and a CNN-based lightweight object detector on our nanocomputer. As illustrated in figure 1, the camera continually captures images (frames) of the environment in front of the user. The captured frames are transmitted to the object detector module, which processes them and infers the nature and the localization of the detected objects. The text-to-speech module is then triggered to provide audio information about obstacles and their nature.

[1] In some challenges such as COCO there is no distinction between "Average Precision (AP)" and "mean Average Precision (mAP)"

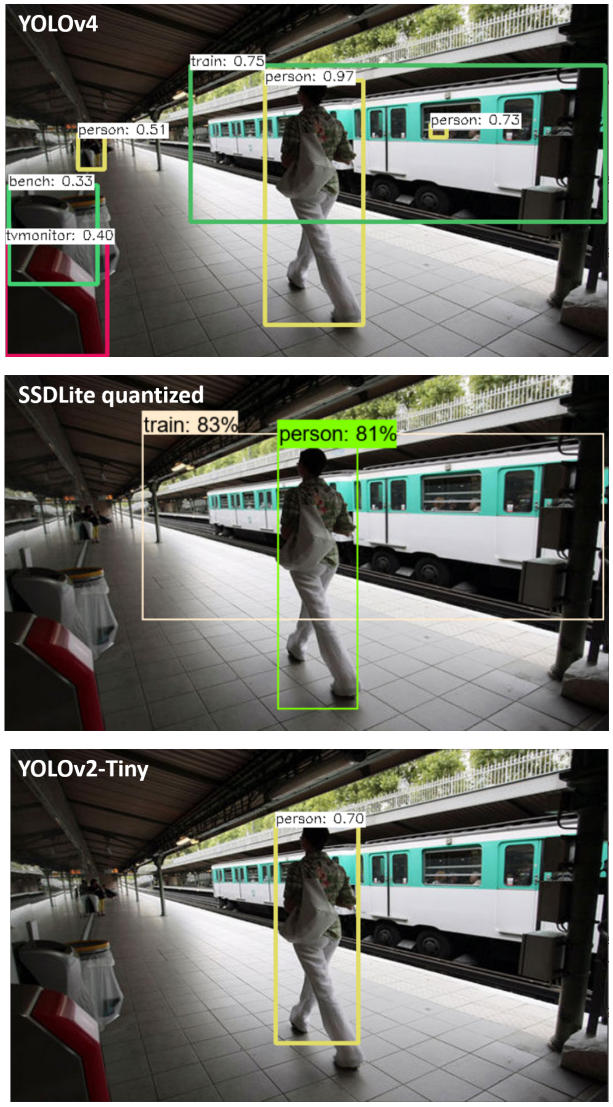


Fig. 3. Differences in detection between the most accurate tested model (YOLOv4), the least accurate tested model (YOLOv2-tiny) and one of the models that offers a good speed-accuracy trade-off (SSDLite MobileNetv2 quantized).

As mentioned before, in most existing assistive systems with full recognition capabilities, both the learning AND the inference processes are run by GPU-based computers/ servers. For each image captured by the device, we need to stream it to the server and back. Such an approach cannot satisfy strict end-to-end, low-latency requirements needed for real-time and interactive applications. In our work, the learning and the inference processes are decoupled. As the computational requirements of the learning process exceed by far the capacity of an embedded system, training is kept off-line. However, in contrast with most of existing systems, we transform the obtained off-line model into an optimized lightweight detector. The lightweight detector is then deployed on the device for a real-time on-device inference ($i.e.$, data locality). The flowchart of inference and training processes are shown in 1. As illustrated in figure 2, our system takes the form of a camera harness embedding a nanocomputer.

As lightweight detectors are inherently less accurate than regular ones, we implemented and conducted an extensive study on state-of-the-art fast detectors to strike the best trade-off between accuracy and inference speed.

B. System implementation

Our system was developed using python and OpenCV 4.5.1. The `cv2.VideoCapture()` module was used to capture the video stream from the Raspberry camera. The text-to-speech module was developed using pyttsx3 [20] which, in contrast with alternative libraries, works entirely offline.

We selected 6 YOLO models: YOLOv2, YOLOv3, YOLOv4 and their corresponding tiny variants. We also selected 3 SSD-MobileNet models and then generated the corresponding quantized variants. The AP@0.5:0.95 accuracy of the selected models ranges from 5.9% to 44.4% and the FPS (Frame per Second) from $\approx$ 0.17 to $\approx$ 11.39 (figure 5). Figure 3 illustrates the differences in detection between the most and the least accurate tested model and one of the tested models that offers a good speed-accuracy trade-off.

1) SSD Models: The evaluation of SSD models was done using both TensorFlow and TensorFlow Lite frameworks without any accelerator hardware, CPU overclocking or special tuning. Inferencing was carried out with SSD MobileNetv1, SSD MobileNetv2, and SSDLite MobileNetv2 models, all trained on the Common Objects in Context (COCO) dataset. Pretrained SSD models were downloaded from the TensorFlow detection model zoo [21]. Tensorflow Lite models were generated using the TFLiteConverter API, which allows to generate frozen graphs (tflite_graph.pb) and to convert frozen graphs to the Tensorflow Lite flatbuffer format (detect.tflite).

Model optimization techniques (called also neural network compression) aim to build models with reduced memory footprint, latency, size, and energy consumption, at the cost of the least possible accuracy degradation. Although optional for models running on GPU-based servers, model optimization techniques are mandatory for on-device models.

At the current stage of our work, we experimented the "post-training 8-bit quantization", which is an efficient compression technique that requires neither specialized hardware nor retraining. Quantization compresses a model by lowering the number of bits required to represent its parameters ($e.g.$ weights and activation outputs). In our quantized models (generated using TensorFlow Lite post-training quantization tools), 32-bit floating-point numbers are converted to the nearest 8 bit integers. As shown in table II, the obtained models are on average ≈ 4 times smaller than the corresponding baseline models.

2) YOLO Models: Choosing the best-suited inference framework is crucial for real-time object detection applications running on devices with limited resources. Native Darknet was designed to run on GPUs and is relatively slow on CPUs [22]. In our work, we used the OpenCV Deep Neural Network inference engine (OpenCV DNN) instead. In addition

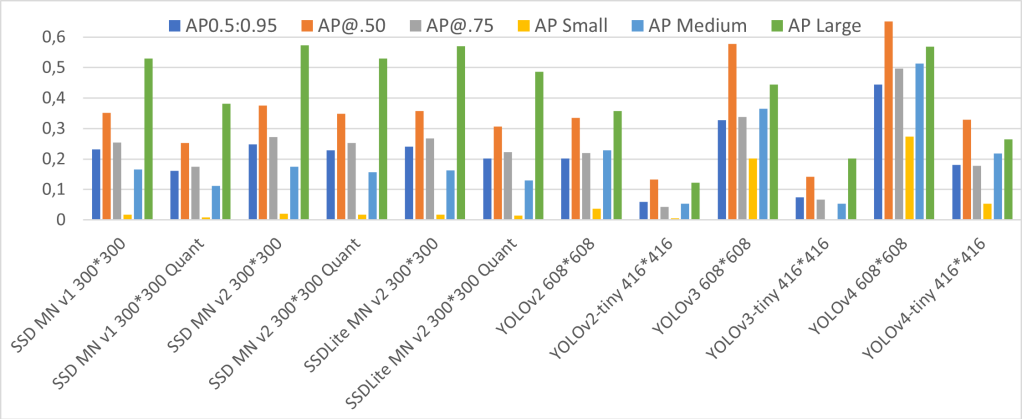


Fig. 4. Comparison of the accuracy of the studied object detectors.

TABLE II
SSD MODELS SIZE COMPARISON.

Model	TF model (frozen inference graph)	Generated TF Lite quantized model (detect.tflite)
SSD MobileNetv1	27.7 MB	6.59 MB
SSD MobileNetv2	66.4 MB	16.1 MB
SSDLite MobileNetv2	18.9 MB	4.43 MB

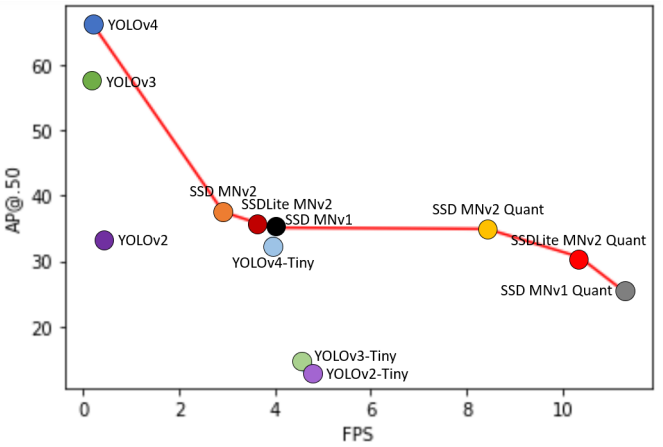


Fig. 5. Speed/accuracy tradeoffs (the Pareto frontier is plotted in red).

to its efficient C/C++ implementation, OpenCV DNN uses several optimizations to speed up inference with little or no loss in accuracy (*e.g.* layer fusion, pruning, etc.). YOLOv4 authors stated that OpenCV DNN is "the fastest inference implementation of YOLO on CPU devices" [23].

YOLO models, pretrained on MS COCO (graph and weights), were loaded in OpenCV using the `cv2.dnn.readNetFromDarknet()` function. The `cv2.dnn.NMSBoxes()` function was used to remove redundant overlapping bounding boxes. We set the DNN backend to OpenCV (`cv2.dnn.DNN_BACKEND_OPENCV`) and the target to CPU (`cv2.dnn.DNN_TARGET_CPU`). The `getUnconnectedOutLayers()` function was used to identify the output layer(s) of the network, which is essential to run the inference function `forward()`. We should notice that OpenCV's DNN module doesn't support quantized models at the time of writing this paper.

C. Experimental study

Investigating Speed/Accuracy trade-offs is of great interest for environments with limited resources [24]. In our work, we conducted two sets of experiments to evaluate the impact of model optimization on accuracy and on inference time.

YOLO and Tensorflow researchers released several easy-to-run pre-trained models and provided their respective AP and FPS [21], [25]. Despite these efforts, there are several reasons that make it difficult for practitioners to decide which model is best suited to their applications [24]. First, the claimed accuracy of YOLO and SSD models are obtained on different datasets (*e.g.* COCO 2014 minival set for SSD models and COCO test-dev set for YOLO models). Second, some important metrics are often omitted (*e.g.* the claimed AP of YOLOv3-tiny is an AP@.5, while the claimed AP of SSD models are AP@[.50:.05:.95]). Finally, except for some SSD models, all reported FPS are obtained on GPUs or using neural sticks, and no indication is given on inference time on CPUs.

To provide a fair comparison between regular and optimized models, we evaluate their accuracy over the COCO val2017 dataset (5k images) using the pycocotools API [17]. For inference speed benchmarking, to get close to the actual use conditions of our system, we considered 10 videos of 2 minutes each, recorded with our camera and involving different indoor and outdoor environments in normal stability/walking conditions (constant and natural walking speed). Reported FPS are average of three consecutive runs and are only based on model run time (*i.e.* `forward()`, `tf.session.run()` and `tf.lite.Interpreter.invoke()`). Preprocessing of an image is not taken into account, nor text-to-speech conversion. Figure 4 depicts the overall AP, the AP for large, medium and small objects on the COCO 2017 validation dataset. Figure 5 presents an exploration of the space of speed/accuracy trade-offs.

As shown in figure 5, the lightweight variants of YOLO are much faster than regular ones: in our experiments, YOLOv2-tiny, YOLOv3-tiny and YOLOv4-Tiny are respectively 10.46,

26.61 and 21.13 times faster than YOLOv2, YOLOv3 and YOLOv4. As shown in figure 4, this inference speed-up is accompanied by a significant drop in accuracy: YOLOv2-tiny, YOLO-V3-tiny and YOLOv4-tiny are respectively 3.42, 4.43 and 2.47 times less accurate than the corresponding regular models. Among the aforementioned tiny models, only YOLOv4-tiny manages to keep a decent overall AP.

In contrast with regular YOLO models, baseline SSD-mobilenet models are, in general, fast enough to run on a low-end device such as the Raspberry Pi 4 (figure5). As shown in figure 5, using Tensorflow Lite in conjunction with post-training quantization allows a substantial inferencing speed-up with a relatively small loss of accuracy. To give an order of magnitude, in our experiments Tensorflow Lite used in conjunction with post-training integer quantization, decreases inferencing time by a factor of 2.86 with an accuracy drop ranging from 3.9% to 7.1%. According to our experiments, SSD MobileNetv1 quantized and SSDLite MobileNetv2 quantized offer both good speed-accuracy trade-offs. The latter was retained as the object detector of our assistive device.

IV. Conclusion

This paper presents a CNN-based assistive device, intended for visually impaired people. Although our system uses off-the-shelf, low-cost materials, it achieves a real-time and accurate object recognition.

In contrast with most of existing CNN-based assistive devices, our system adopts an on-device inference approach. On-device inference offers a variety of benefits: lower inference time (by removing the need to stream data to the cloud and back); increased security and privacy; increased reliability (by removing the need for a fast and active internet connection); and reduced costs.

Running state-of-the-art object detectors on devices with constrained resources is challenging and often prohibitive due to their high computational requirements. The approach adopted in our work is to consider lightweight variants of existing models and apply compression techniques to reduce model size and speed up inferencing time. Besides, we performed an extensive experimental comparison of state-of-the-art lightweight object detectors to identify models and frameworks that achieve the best speed/accuracy trade-offs. We hope this will help choosing the appropriate model when deploying object detection on devices with limited resources.

As a part of our future work, we intend to exploit temporal information from videos to predict movements and avoid eventual collisions.

References

[1] World Health Organization, "Blindness and vision impairment," https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment, 2019, [Accessed: 2020-06-01].

[2] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *CoRR*, vol. abs/1704.04861, 2017. [Online]. Available: http://arxiv.org/abs/1704.04861

[3] J. Redmond, S. Divvala, R. Girshick, and A. Farhadi, "Unified, real-time object detection," *CoRR*, vol. abs/1505.06798, 2017.

[4] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6517–6525.

[5] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv*, 2018.

[6] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," 2020.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[8] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[9] J. Chen and X. Ran, "Deep learning with edge computing: A review," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1655–1674, 2019.

[10] S. Ravi, "Projectionnet: Learning efficient on-device deep networks using neural projections," *arxiv*, 2017. [Online]. Available: https://arxiv.org/pdf/1708.00630.pdf

[11] B. Lin, C. Lee, and P. Chiang, "Simple smartphone-based guiding system for visually impaired people," *Sensors*, vol. 17, no. 6, p. 1371, 2017. [Online]. Available: https://doi.org/10.3390/s17061371

[12] B. Mocanu, R. Tapu, and T. Zaharia, "Seeing without sight : An automatic cognition system dedicated to blind and visually impaired people," in *2017 IEEE International Conference on Computer Vision Workshops, ICCV Workshops 2017*, 2017, pp. 1452–1459.

[13] ——, "Design of a CNN face recognition system dedicated to blinds," in *IEEE International Conference on Consumer Electronics, ICCE 2019, Las Vegas, NV, USA, January 11-13, 2019*. IEEE, 2019, pp. 1–2. [Online]. Available: https://doi.org/10.1109/ICCE.2019.8661933

[14] F. Rahman, I. J. Ritun, N. Farhin, and J. Uddin, "An assistive model for visually impaired people using yolo and mtcnn," in *Proceedings of the 3rd International Conference on Cryptography, Security and Privacy*, ser. ICCSP '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 225–230.

[15] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *CoRR*, vol. abs/1311.2524, 2013. [Online]. Available: http://arxiv.org/abs/1311.2524

[16] C. Wang, H. M. Liao, Y. Wu, P. Chen, J. Hsieh, and I. Yeh, "Cspnet: A new backbone that can enhance learning capability of CNN," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2020, Seattle, WA, USA, June 14-19, 2020*. IEEE, 2020, pp. 1571–1580.

[17] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014. [Online]. Available: http://arxiv.org/abs/1405.0312

[18] R. Hadidi, J. Cao, Y. Xie, B. Asgari, T. Krishna, and H. Kim, "Characterizing the deployment of deep neural networks on commercial edge devices," in *2019 IEEE International Symposium on Workload Characterization (IISWC)*, 2019, pp. 35–48.

[19] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[20] pyttsx3, https://pypi.org/project/pyttsx3/, [Accessed: 2020-06-01].

[21] TensorFlow 1 Detection Model Zoo, https://github.com/tensorflow/models/, [Accessed: 2020-06-01].

[22] Y. Koo, S. Kim, and Y.-g. Ha, "Opencl-darknet: implementation and optimization of opencl-based deep learning object detection framework," *World Wide Web*, pp. 1–21, 02 2020.

[23] Alexey Bochkovskiy, https://github.com/AlexeyAB/darknet, [Accessed: 2021-01-01].

[24] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, "Speed/accuracy trade-offs for modern convolutional object detectors," *CoRR*, vol. abs/1505.06798, 2017.

[25] YOLO: Real-Time Object Detection, https://pjreddie.com/darknet/yolo/, [Accessed: 2020-06-01].

.

2. The Database Version Approach : Overview and Future directions

Talel Abdessalem, Claudia Medeiros Bauzer, Wojciech Cellary, Stéphane Gançarski, Khaled Jouini, Maude Manouvrier, Marta Rukoz, Michel Zam

34ème Conférence sur la Gestion de Données – Principes, Technologies et Applications (BDA). Romania. 2018.

Lien : https://shs.hal.science/BDA2018/hal-02191121v1

The Database Version Approach: Overview and Potential Directions

In tribute to Geneviève Jomier (1948 - 2018)

Talel Abdessalem
LTCI, Télécom ParisTech, Université Paris-Saclay
Paris, France
talel.abdessalem@telecom-paristech.fr

Claudia Bauzer Medeiros
University of Campinas, Institute of Computing (IC - UNICAMP)
Campinas, Brazil
cmbm@ic.unicamp.br

Wojciech Cellary
Poznan University of Economics
Poznan, Poland
cellary@kti.ue.poznan.pl

Stéphane Gançarski
LIP6 - U.P.M.C
Paris, France
Stephane.Gancarski@lip6.fr

Khaled Jouini
MARS Research Lab LR17ES05, ISITCom, University of Sousse
Sousse, Tunisia
khaled.jouini@isitc.u-sousse.tn

Maude Manouvrier
Université Paris-Dauphine, PSL Research University, CNRS UMR [7243] LAMSADE
Paris, France
maude.manouvrier@dauphine.fr

Marta Rukoz
Université Paris-Dauphine, PSL Research University, CNRS UMR [7243] LAMSADE
Paris, France
marta.rukoz@dauphine.fr

Michel Zam
Université Paris-Dauphine, PSL Research University, CNRS UMR [7243] LAMSADE
Paris, France
zam@dauphine.fr

ABSTRACT

In 1990, W. Cellary and G. Jomier proposed the Database Version (DBV) approach, which allows to manage multiversion databases — those in which several versions of a set of data items coexist. Ever since, its model, theory and algorithms have been adopted in a multitude of research initiatives and publications, and been applied to a variety of applications, in particular those in which there is a need for keeping track of parallel or (spatio)-temporal evolution of states of the world. This article presents an overview of the DBV approach, and some of the associated research initiatives throughout three decades, pointing out new potential directions. It has been written in tribute to Geneviève Jomier, Prof. Emeritus of The Université of Paris-Dauphine, who left us in March 2018.

1 INTRODUCTION

This article presents the *Database Versions* (DBV) approach, proposed by W. Cellary and G. Jomier in 1990 in [10].

Formally, a database is monoversion, representing only one state of the modeled world; each data item has a single value. In the DBV approach, a multiversion database brings together several states of the world, these states being variants or evolutions in time of the modeled world. Each data item has, in this case, several versions, the value of that item potentially changing from one version to another. Each Database Version represents a consistent state or a configuration of the modeled world [17]. DBVs can be applied to any type of data, whether images [35, 36], documents [5] or spatio-temporal data [42, 43, 48].

As will be seen throughout this text, the DBV model can be adopted in a variety of situations, whenever there is a need for managing many states of the world. Though this may seem to be an obvious statement, since it models versions (and thus states of the world), it is simple and generic. Through the DBV model, versions can be managed through a small set of compact data structures, simplifying maintenance of co-existing states, and speeding up rollback to any database state. For this reason, it was implemented in many computing platforms, for several purposes, in a variety of contexts.

The DBV model has originated several research initiatives, including the ones presented in a variety of papers - e.g., [2, 5, 17, 24, 32, 33, 36–39, 48, 49, 52]. Several PhD thesis in computer science [1, 18, 22, 35, 42, 50], supervised by G. Jomier, are based on this model. The model has also inspired thesis [41] and research projects - e.g., [13, 31, 44], in other institutions.

Section 2 presents the DBV approach and Section 3 discusses a few of its applications. Section 4 concludes the paper, presenting new potential research directions which can still be exploited.

2 DBV APPROACH

This section gives an overview of the DBV approach and of its subsequent generalizations.

2.1 Overview of the concepts

A DBV is identified by v. A multiversion data is associated with an immutable identifier, d. A DBV contains exactly one *logical version* d_j of each multiversion data d, having an identifier and a value val.

.

3 Fusion de minuties pour une reconnaissance efficiente des empreintes digitales

Mohamed Hédi Ghaddab, Khaled Jouini & Ouajdi Korbaa
Colloque sur l'Optimisation et les Systèmes d'Information (COSI). Algérie. 2017.

Lien : https://cosi.isima.fr/cosi2017/ArticlesLongsAcceptes.html

Fusion de minuties pour une reconnaissance efficiente des empreintes digitales

Mohamed Hédi Ghaddab[1], Khaled Jouini[2], and Ouajdi Korbaa[3]

[1] ghaddab.mohamedhedi@gmail.com
[2] j.khaled@gmail.com
[3] Ouajdi.Korbaa@centraliens-lille.org
Laboratoire MARS* LR17ES05
Institut Supérieur d'Informatique et des Techniques de Communication (ISITCom)
Université de Sousse, Tunisie

Résumé La reconnaissance efficiente des empreintes digitales est tributaire de la qualité des empreintes de référence (pré-stockées), auxquelles sont comparées les empreintes requêtes. Une des techniques classiques pour améliorer la qualité des empreintes de référence est de fusionner plusieurs impressions d'une même empreinte en une seule.

Une empreinte se caractérise par un ensemble de points, dits *minuties* correspondant à la terminaison ou à la bifurcation de ses crêtes. Du fait de leur haut pouvoir discriminant, les minuties sont à la base de la majorité des algorithmes de comparaison d'empreintes.

Cet article propose une nouvelle approche de fusion, intitulée FZC. FZC se distingue de la majorité des approches existantes par (*i*) l'application d'alignements différents sur différentes régions de l'empreinte ; et (*ii*) une estimation de la crédibilité d'une minutie qui ne se base pas sur sa fréquence dans les différentes impressions, mais sur la qualité de la capture de la région dans laquelle la minutie se trouve. Les résultats expérimentaux obtenus confirment le bien-fondé de l'approche proposée.

Keywords: Comparaison d'empreintes ; Super-modèle ; Fusion de minuties.

1 Introduction

Le traitement automatisé d'empreintes digitales est au cœur de nombre d'applications sensibles, des plus classiques telles que celles ayant trait à la lutte contre la criminalité, aux plus récentes comme la sécurité d'accès aux données dans le contexte des réseaux ouverts. Le traitement automatisé des empreintes digitales suit sommairement les étapes suivantes : acquisition, extraction de caractéristiques et de descripteurs, stockage et comparaison (*i.e.* vérification) ou recherche (*i.e.* identification). Dans la suite nous nous intéressons à la comparaison automatisée d'empreintes digitales.

Une empreinte digitale se caractérise par un ensemble de crêtes et de points singuliers, dits *minuties* correspondant à la terminaison ou à la bifurcation des crêtes. Une minutie m est communément décrite par un quadruplet $m =$

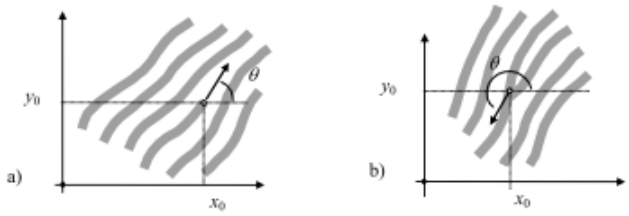


Figure 1. Grands types de minuties : (a) terminaison ; (b) Bifurcation. [11].

(x, y, θ, t), où (x, y) correspondent à sa localisation spatiale, θ à l'angle formé par l'axe horizontal et la tangente à la ligne de crête au point de minutie (figure 1) et t à son type (bifurcation ou terminaison). Bien qu'il existe différentes approches pour la comparaison d'empreintes digitales, celle basée sur les points de minuties demeure largement la plus répandue [13]. La comparaison de deux empreintes basée sur leurs minuties revient sommairement à superposer les minuties (*i.e.* alignement) et à calculer un score en fonction des minuties concordantes.

Pour une même empreinte, il est extrêmement difficile que deux captures différentes aboutissent à une même représentation. Cette variabilité intra-classe s'explique par des propriétés intrinsèques et extrinsèques de l'empreinte. Les différentes techniques de capture d'empreintes ont pour propriété commune de transformer l'objet tridimensionnel qu'est l'empreinte en une image bidimensionnelle. Cette perte de dimension, conjuguée à des propriétés intrinsèques de l'empreinte comme l'élasticité de la peau et la différence entre les forces appliquées lors de l'apposition du doigt, fait qu'il soit extrêmement difficile que des captures différentes d'une même empreinte aboutissent à une localisation spatiale identique des minuties. Les propriétés extrinsèques ont trait aux erreurs introduites lors de l'extraction de minuties. Dans le cas d'images d'empreintes de mauvaise qualité ou d'empreintes obtenues par encrage du doigt, les algorithmes de détection de minuties peuvent introduire un grand nombre de fausses minuties (*i.e. spurious minutiae*) et ne pas être en mesure de détecter toutes les vraies minuties (*i.e. missing minutiae*). Les minuties déplacées, manquantes ou fausses sont la principale source de faux rejets et d'erreurs de vérification.

La qualité des empreintes de référence pré-stockées, auxquelles sont comparées les empreintes requêtes, a une incidence importante sur l'efficience de tout système de reconnaissance d'empreintes. Une des techniques classiques pour améliorer la qualité des empreintes de référence est de fusionner plusieurs impressions d'une même empreinte en une seule. Sans perte de généralité, nous utilisons dans la suite le terme modèle (*template*) pour désigner l'ensemble de minuties extraites d'une impression d'empreinte. La fusion de plusieurs modèles d'une même empreinte en un seul modèle, dit *super-modèle* (*super template*), a deux grands objectifs : la consolidation du modèle de référence (*template consolidation*) et l'adaptation du modèle de référence aux variations que peut subir une empreinte au fil du temps suite par exemple à des coupures ou des cicatrices causées par des blessures (*template learning* ou *template adaptation*) [17].

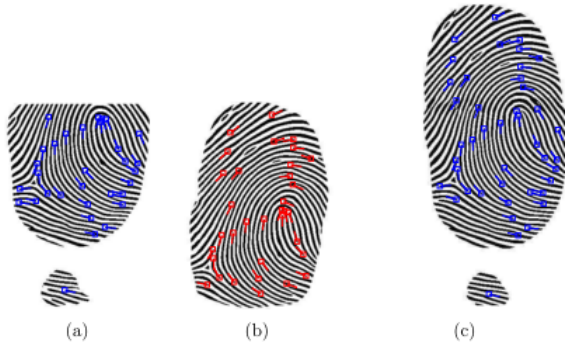


Figure 2. Fusion au niveau capteur : (a) Impression 1. (b) Impression 2. (c) Image composite [15]

Cet article propose une nouvelle approche pour la construction de super-modèles, intitulée FZC. Les difficultés majeures rencontrées lors de la construction d'un super-modèle sont l'estimation de la "crédibilité" d'une minutie et l'alignement des impressions à fusionner lorsque ceux-ci sont fortement affectés par les distorsions. L'approche que nous proposons part du constat que les distorsions et les déformations n'affectent pas d'une manière uniforme les différentes régions d'une empreinte. Elle se distingue de la majorité des approches existantes par l'application d'alignements différents sur différentes régions de l'empreinte. L'approche proposée se distingue également par une estimation de la crédibilité d'une minutie qui ne se base pas sur sa fréquence dans les différentes impressions, mais sur la qualité de la capture de la région dans laquelle la minutie se trouve. Les résultats expérimentaux obtenus sur les bancs de test standards, confirment le bien-fondé de l'approche proposée.

Dans la suite de l'article, la section 2 donne une brève revue des principales approches de fusion d'empreintes. La section 3 présente l'approche FZC. La section 4 étudie expérimentalement les performances de FZC. Suit la conclusion.

2 Principales approches de fusion d'impressions d'une empreinte

En biométrie le terme "fusion au niveau capteur" (*Sensor-Level*) est utilisé pour désigner la combinaison de données biométriques brutes pour former une caractéristique biométrique composite. Le terme "fusion au niveau caractéristiques" (*Feature-Level*) est communément utilisé pour désigner la combinaison des vecteurs de caractéristiques obtenus de différentes sources (différents capteurs, échantillons ou modalités biométriques). La fusion à ce niveau pré-traite les données brutes pour en extraire des descripteurs, puis combine les descripteurs extraits en un seul.

Dans le cas précis des empreintes digitales, la fusion au niveau du capteur se matérialise le plus souvent par la fusion d'images correspondant à différentes

captures d'une même empreinte. La fusion des caractéristiques se matérialise le plus souvent par la combinaison de minuties extraites de différentes impressions. Dans la suite ce cette section nous présentons les principales approches de fusion d'images et de minuties d'une même empreinte [4].

2.1 Fusion d'images

Jain et Ross [7] ont proposé une approche de "mosaïquage" (*mosaicking*) fusionnant les différentes images d'une empreinte en une seule (voir figure 2). L'approche proposée utilise un algorithme de comparaison basé sur les minuties pour trouver le bon alignement entre les images à fusionner. Les images sont ensuite rendues compatibles en normalisant les contrastes et les intensités des pixels. L'image finale est obtenue par la superposition de toutes les (sous-)images alignées, lissées et normalisées.

Plusieurs variantes de la technique de mosaïquage ont été proposées [12,3,19,4]. La plupart de ces approches diffèrent soit dans la manière d'aligner et d'harmoniser les images soit dans le domaine d'application du mosaïquage. À titre d'exemple, dans [3] les auteurs s'intéressent aux capteurs à petites surfaces et ont pour objectif d'augmenter la surface de l'empreinte couverte par l'image de référence. Les auteurs recommandent la capture de différentes parties de l'empreinte et d'utiliser le mosaïquage ensuite pour recomposer une image complète de l'empreinte. Zhang et al. [19] appliquent la technique de mosaïquage aux capteurs à balayage d'empreintes. L'approche consiste à fusionner les images obtenues par des balayages différents d'une empreinte pour améliorer la qualité de l'image de référence destinée à être stockée.

Pour des raisons de confidentialité et de maîtrise du coût du stockage, plusieurs systèmes de reconnaissance ne stockent pas les images d'empreintes et se contente d'enregistrer les descripteurs qu'ils en extraient. Ceci constitue une limitation importante de la technique de fusion d'images. Par ailleurs, les différentes études menées pour comparer la fusion d'images et la fusion de caractéristiques, telles que celle de [15] et de [5], ont montré que bien que les deux approches améliorent la précision de la reconnaissance d'empreintes, la fusion de caractéristiques surclasse la fusion d'images.

2.2 Fusion de minuties

La fusion de minuties extraites d'impressions différentes d'une même empreinte a pour objectif la construction d'un super-modèle (*template synthesis*), plus distinctif et plus facile à comparer que les impressions prises individuellement. Yau et al. [18] partent de l'hypothèse que le module d'extraction de minuties n'introduit que peu de fausses minuties, mais qu'il est le plus souvent incapable de détecter toutes les vraies minuties. L'approche proposée aligne les

4. Il existe une troisième alternative, dite "fusion des scores", consistant à appliquer différents algorithmes de comparaison, puis à normaliser et combiner les différents scores en un seul. La fusion de scores sort du cadre de cet article.

différentes impressions d'une empreinte, puis fait l'union des minuties extraites de chaque impression. Au fur et à mesure de l'union, les attributs (localisation spatiale, angle d'orientation et type) des minuties communes à différentes impressions sont corrigés. L'approche proposée par [18] améliore donc la tolérance aux minuties déplacées et manquantes, mais reste vulnérable aux minuties parasites. L'approche ne permet pas non plus l'adaptation de la représentation de référence aux variations que peut subir une empreinte au fil du temps.

Ramoser et al. [14] proposent une approche similaire à celle de Yau et al. [18]. Elle se distingue cependant par une méthode originale d'alignement des impressions basée sur l'algorithme RANSAC [5]. Tout comme l'approche proposée dans [18], celle proposée dans [14] ne permet pas la suppression de fausses minuties et l'adaptation continue de la représentation de référence d'une empreinte.

Jiang et Ser [8] ont proposé une extension à l'approche de Yau et al. [18] permettant la prise en compte de la fiabilité d'une minutie et l'adaptation continue du modèle de référence d'une empreinte. Dans [8] les impressions sont collectées au fur et à mesure que des empreintes requêtes sont soumises au système (adaptation en ligne). Plus une minutie se répète dans les impressions collectées d'une empreinte, plus elle est jugée fiable. Chaque fois qu'une impression de référence est comparée avec une impression requête de la même source, la fiabilité associée aux minuties communes est incrémentée de un dans le modèle de référence. Au fil de la collecte des impressions, les vraies minuties finissent par obtenir les plus hauts scores de fiabilité. Lors de la comparaison d'empreintes, les minuties aux plus bas scores sont considérées comme parasites et sont ignorées.

Jiang et Ser [8] proposent également de corriger les attributs d'une minutie, au fur et à mesure de la collecte des impressions. La valeur donnée à un attribut est ainsi la moyenne pondérée des valeurs observées de cet attribut dans les différents échantillons. Comme le reporte les auteurs, ces différentes améliorations conduisent à : (*i*) l'élimination progressive des minuties parasites ; (*ii*) la correction des valeurs d'attribut des minuties ; et (*iii*) à une plus grande précision dans la comparaison d'empreintes. Cette approche n'est cependant probante que lorsque le nombre d'échantillons collectés par empreinte est significatif : 2 ou 3 impressions ne sont pas suffisantes pour aboutir à des corrections ou à des scores de fiabilité significatifs.

L'idée d'associer un score de fiabilité aux minuties en fonction de leurs nombres d'occurrences dans différents échantillons, a été reprise par Uz et al. [17] dans leur approche intitulée comparaison hiérarchique (*i.e. Hierarchical Matching*). Contrairement à Jiang et al. [8] qui ignorent les minuties à faibles scores de fiabilité lors de la comparaison d'empreintes, Uz et al. [17] considèrent toutes les minuties détectées et se servent des scores de fiabilité comme coefficients de pondération.

5. RANSAC (*RANdom SAmple Consensus*) est un algorithme itératif qui s'utilise lorsque l'ensemble de données observées est susceptible de contenir des valeurs aberrantes. RANSAC permet de générer un alignement qui ne tient compte que des valeurs pertinentes.

Hierarchical Matching examine les minuties de manière hiérarchique selon leurs scores de fiabilité. Une triangulation de Delaunay différente est construite pour chaque sous ensemble de minuties de même fiabilité. L'objectif de cette représentation hiérarchique est de trouver la meilleure transformation permettant d'aligner une impression donnée à la représentation de référence de l'empreinte. L'alignement de l'échantillon se fait tout d'abord par rapport aux (triangles de) minuties avec le plus haut score de fiabilité (ensemble supposé ne pas contenir de minuties parasites). Cet alignement est raffiné progressivement en considérant tour à tour les (triangles de) minuties de score plus faible.

La comparaison hiérarchique souffre de la même limitation que celle de proposée dans [8] : elle n'est probante que lorsque le nombre d'échantillons collectés par empreinte est significatif. Par ailleurs, elle se base sur la triangulation de Delaunay, connue pour être vulnérable aux minuties manquantes et parasites [11].

3 Alignement de régions compatibles pour la construction de super-modèles

3.1 Vue d'ensemble

Dans cette section nous présentons notre nouvelle approche de construction de super-modèle par fusion de minuties, intitulée FZC (pour Fusion de Zones Compatibles). L'objectif de FZC est d'améliorer la qualité de l'impression de référence d'une empreinte par : (*i*) la restauration des minuties manquantes ; (*ii*) l'élimination des minuties parasites ; et (*iii*) l'augmentation de la surface de l'empreinte couverte par l'impression de référence.

L'approche FZC se distingue de celles existantes en deux points clés : l'estimation de l'authenticité d'une minutie et l'alignement des impressions de l'empreinte. La majorité des approches existantes se basent sur la fréquence (*i.e.* nombre d'occurrences) d'une minutie dans les différentes impressions pour juger de son authenticité. La fréquence d'une minutie peut se révéler insuffisante lorsqu'il n'est pas possible de collecter un nombre significatif d'impressions ou bien lorsque les impressions collectées sont de mauvaise qualité ou partielles. Dans notre approche, l'authenticité d'une minutie est estimée essentiellement par la qualité de la capture de la région dans laquelle se trouve la minutie. Si une minutie se trouve dans une région affectée par une forte distorsion (*e.g.* orientations ou courbures aberrantes des lignes de crêtes) ou bien dans une région obscurcie de l'image, la minutie est jugée peu fiable. Autrement, elle est jugée fiable.

Plusieurs algorithmes et outils permettent d'associer une mesure de qualité aux minuties extraites d'une empreinte [6,2,16]. Cette mesure de qualité est considérée comme donnée dans notre approche. Nous représentons donc une minutie m par un quadruplet $m = (x, y, \theta, t, Q)$, où Q correspond à la mesure de qualité normalisée fournie par l'algorithme d'extraction de minuties.

La première impression d'une empreinte est dite impression ou modèle de référence. Chaque fois qu'une nouvelle impression de la même empreinte est disponible (lors de l'enregistrement ou ultérieurement), les minuties de l'impression

en entrée sont fusionnées avec les minuties de l'impression de référence. Le résultat de cette fusion est un nouveau modèle de référence, dit super-modèle, synthétisant l'ancien modèle de référence et le modèle en entrée. Le modèle de référence et le modèle en entrée sont désignés dans la suite respectivement par Ref et I. La fusion des minuties de Ref et I se passe en deux étapes : (i) appariement des minuties et détermination des régions locales compatibles ; et (ii) alignement des minuties de chaque paire de régions compatibles et consolidation.

3.2 Régions locales compatibles

L'appariement de Ref et I donne lieu à trois groupes de minuties : (i) les minuties de Ref qui n'ont pas de correspondant dans I ; (ii) les minuties de I qui n'ont pas de correspondant dans Ref ; et (iii) l'ensemble noté $M = \{(p_i^R, q_j^I) \mid p_i^R \in Ref \ , \ q_j^I \in I\}$ de paires de minuties de Ref et I qui concordent.

Sommairement, l'idée générale derrière FZC est de trouver les paires de régions locales de I et Ref qui correspondent selon les paires de M. Ces régions sont dites *régions compatibles*. Ensuite, pour chaque paire de régions compatibles, nous cherchons l'alignement local permettant d'inclure dans le modèle final, les minuties de I qui n'ont pas de correspondant dans Ref.

Une région locale se définit par une minutie centrale et un certain nombre de minuties qui se trouvent aux-alentours. Nous considérons que deux régions de Ref et I sont compatibles (correspondent), si elles contiennent un nombre Th_n de minuties appariées dans un rayon inférieur à Th_c. Une paire Z_i de régions compatibles se définit par

$$Z_i = (center_i^R, center_i^I, M_i) \tag{1}$$

où $center_i^R$ est la minutie centrale de la région dans l'impression Ref, $center_i^I$ la minutie centrale de la région correspondante dans I et M_i l'ensemble de paires de minuties des deux régions qui concordent.

Nous utilisons l'algorithme 1 (présenté ci-après) pour déterminer les régions compatibles de Ref et I. Cet algorithme est inspiré de l'algorithme de partitionnement en K-moyennes (*i.e.* K-means). Il construit progressivement des groupes (*clusters*) de Th_n minuties se trouvant à une distance de moins de Th_c d'une minutie centrale, qui s'ajuste chaque fois qu'une minutie est ajoutée au groupe. Une minutie n'est ajoutée à un groupe donné, que si elle apparaît à la fois dans I et Ref. Les groupes construits ainsi, sont nos régions compatibles.

Les figures 3.a, 3.b et 3.c donnent une illustration de l'application de l'algorithme 1 sur deux impressions réelles d'une même empreinte. Les ellipses de la figure 3.c correspondent aux régions compatibles obtenues à partir des impressions 3.a et 3.b. Les régions compatibles contiennent des minuties de 3.a et 3.b qui se superposent.

3.3 Alignement et fusion des régions compatibles

Comme indiqué précédemment, FZC part des constats que : (i) les distorsions n'affectent pas de manière uniforme les différentes régions d'une empreinte ; (ii)

Entrées : Ensemble M de paires de minuties de Ref et I qui se superposent
Sorties : Ensemble de régions compatibles Z

foreach *Paire* (p_i, q_j) *in* M **do**
  if (p_i, q_j) *n'a pas été traitée* **then**
    créer une nouvelle région Z_k ;
    $center_k^R = p_i$;$center_k^I = q_j$;
    Ajouter (p_i, q_j) à M_k ;
    (p_i, q_j) est marquée traitée ;
    repeat
      (p'_i, q'_j) un autre paire de M non traitée ;
      if $distance(center_k^R, p'_i) < Th_c$ *ET* $distance(center_k^I, q'_j) < Th_c$ **then**
        Ajuster les centres ;
        Ajouter (p'_i, q'_j) à M_k ;
        Marquer (p'_i, q'_j) comme traitée ;
      end
    until *Nombre de paires dans* $M_k < Th_n$;
    Ajouter z_k à Z ;
  end
end

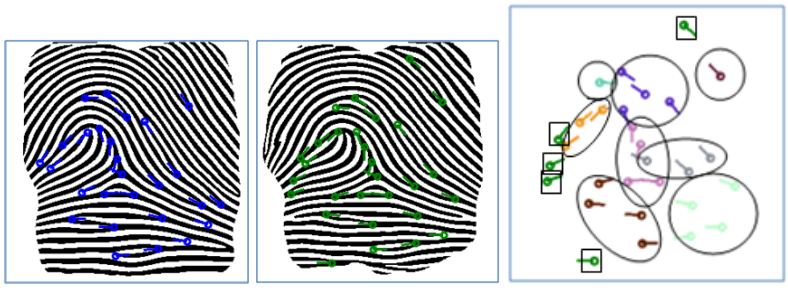
Algorithm 1: Algorithme de partitionnement des impressions en régions locales compatibles.

les distorsions affectent moins les régions locales d'une empreinte que l'empreinte elle-même prise à un niveau globale.

La majorité des algorithmes de comparaison basés sur les minuties finissent à l'étape de consolidation par appliquer un alignement global pour apparier les minuties [13]. Cet alignement global, aussi bon soit-il, n'est pas en mesure d'aligner toutes les minuties qui devraient l'être, compte tenu du fait que des régions différentes ont des distorsions différentes et nécessitent donc des alignements différents. Contrairement à la majorité des approches existantes, FZC procède à des alignements différents pour les régions compatibles.

Le but de l'alignement d'une paire de régions compatibles $Z_i = (z_{Ref}, z_I)$ est de trouver une transformation locale plus fine que celle globale appliquée par l'algorithme de comparaison. Les paramètres de la transformation locale (rotation et translation) sont déduits de la différence entre les localisations spatiales de la minutie $p \in z_{Ref}$ ayant la plus haute fiabilité Q et la minutie correspondante $q \in z_I$. Une fois z_{Ref} et z_I finement alignées et que les localisations spatiales des minuties de z_I sont corrigées en fonction de la transformation locale qui en résulte, les opérations suivantes ont lieu :

1. Si une minutie q_i de z_I est appariée à une minutie p_i de z_{Ref} (*i.e.* $(p_i, q_i) \in M_i$) et que q_i est de meilleure qualité que p_i, alors les attributs de q_i remplacent ceux de p_i dans le modèle de référence. Si p_i et q_i ont des qualités proches à un certain degré, la valeur de chaque attribut de p_i est remplacée par la moyenne pondérée par la qualité, des valeurs de l'attribut dans p_i et q_i.



(a) (b) (c)

Figure 3. (a) Minuties extraites de l'impression 12_6 de la base $FVC2000\ DB1\_A$. (b) Minuties extraites de l'impression 12_3 de la même empreinte. (c) Régions compatibles et minuties sans correspondances.

2. Si une minutie q de z_I n'est appariée à aucune minutie de z_{Ref} et que la mesure de qualité qui lui est associée dépasse un certain seuil Th_Q, q est ajoutée au modèle de référence de l'empreinte, contribuant ainsi à la restauration des minuties manquantes.
3. Si une minutie p de z_{Ref} n'est appariée à aucune minutie de z_I et que la mesure de qualité qui lui est associée est en deçà de Th_Q, p est supprimée du modèle de référence.

4 Étude expérimentale

Les bancs de tests FVC sont communément utilisés pour l'évaluation des algorithmes de comparaison d'empreintes [11]. Pour valider notre approche, nous avons conduit des tests sur les quatre bases du FVC2000 [10]. Notre choix pour ces bases s'explique par le fait que les images qu'elles contiennent ont été acquises avec des scanners d'empreintes à petites surfaces de capture, sans contrôle de qualité et sans que la surface du capteur soit systématiquement nettoyée [11]. La qualité des images de la base DB1_A du FVC 2000 est particulièrement mauvaise [11].

Le banc FVC 2000 est composé de 4 bases contenant chacune 8×100 images, où 100 est le nombre d'empreintes différentes représentées dans la base et 8 le nombre d'impressions différentes par empreinte. Le protocole des bancs FVC comprend deux types de tests permettant d'estimer le FMR (*i.e. False Match Rate*) et le FNMR (*i.e. False Non-Match Rate*). La première série de tests appelée *Genuine tests* a pour objectif d'estimer le FNMR et consiste à comparer chaque impression d'une empreinte à toutes les autres impressions de la même empreinte. Le FNMR se calcule par le fraction d'impressions authentiques qui ont obtenu un score de similarité en dessous du seuil de décision. La deuxième série de tests appelée *Impostor tests* a pour objectif d'estimer le FMR et consiste à comparer la première impression d'une empreinte à la première impression de

		Nombre d'échantillons fusionnés			
		1	2	3	4
DB1_A	EER	1,572	0,649	0,622	0,707
	FMR100%	1,893	0,833	0,800	1,000
	FMR1000%	2,893	1,333	1,200	1,500
	ZeroFMR%	3,357	2,167	1,400	4,250
DB2_A	EER	1,122	0,283	0,346	0,347
	FMR100%	1,250	0,333	0,400	0,250
	FMR1000%	1,964	0,833	0,600	0,750
	ZeroFMR%	2,500	1,000	0,600	0,750
DB3_A	EER	4,228	1,831	1,985	2,020
	FMR100%	6,036	2,333	3,000	3,000
	FMR1000%	7,750	3,667	5,000	4,750
	ZeroFMR%	13,464	8,333	7,800	8,000
DB4_A	EER	2,074	0,990	1,084	1,609
	FMR100%	2,714	1,333	1,400	1,750
	FMR1000%	4,893	2,500	2,200	2,000
	ZeroFMR%	9,250	3,667	6,000	3,750

Table 1. Précision (en %) dans les quatres bases de données utilisées dans la compétitions FVC2000

		Nombre d'échantillons fusionnés			
		1	2	3	4
Hierarchical Matching [17]	EER	–	–	–	–
	FMR100%	–	2,810	1,750	1,230
	FMR1000%	–	4,680	3,100	2,030
	ZeroFMR%	–	8,710	5,230	3,900
FZC	EER	1,572	0,649	0,622	0,707
	FMR100%	1,893	0,833	0,800	1,000
	FMR1000%	2,893	1,333	1,200	1,500
	ZeroFMR%	3,357	2,167	1,400	4,250

Table 2. Étude comparative. Précision (en %) dans la base de données DB1_A utilisée dans la compétitions FVC2000

chacune des empreintes restantes. Le FMR se calcule par la fraction d'impressions imposteurs qui ont obtenu un score de similarité au-dessus du seuil de décision.

Nous avons comparé la précision de FZC à l'approche *Hierarchical Matching* proposée dans [17] (voir section 2.2). La bibliothèque libre VeriFinger [1] a été utilisée pour extraire les minuties et leur associer une mesure de qualité. Nous avons testé l'effet de la fusion de k échantillons sur la précision de la comparaison d'empreintes. Nous avons utilisé l'algorithme GMMS [9] pour le choix des k échantillons qui représentent le mieux les variations dans les impressions d'une

empreinte. Pour ne pas biaiser le calcul du FNMR, les k impressions fusionnées de chaque empreinte, sont exclues des tests *genuine*. Le nombre de tests *genuine* est donc égal à $(8-k) \times 100$ tests.

Le tableau 1 reporte les résultats obtenus par notre approche dans les quatre bases du FVC 2000. Le tableau 2 donne les résultats obtenus par notre approche et celle de [17] sur la base DB1_A du FVC 2000. Les résultats de l'approche *Hierarchical Matching* sont ceux reportés dans [17]. Tel que le montre le tableau 2, notre approche présente une nette supériorité par rapport à [17].

Conclusion

Dans ce article nous avons proposé FZC, une nouvelle approche pour la construction d'un super-modèle à partir de différentes impressions d'une empreinte. L'objectif de FZC est d'améliorer la qualité de l'impression de référence d'une empreinte par : (*i*) la restauration des minuties manquantes ; (*ii*) l'élimination des minuties parasites ; et (*iii*) l'augmentation de la surface de l'empreinte que l'impression de référence couvre. FZC se distingue de la majorité des approches existantes par un alignement différent pour chaque région de l'empreinte. FZC se distingue également par une estimation de la fiabilité d'une minutie qui se base sur la qualité de la capture de la région dans laquelle elle se trouve. Ceci permet une meilleure détection des minuties parasites dans les cas où peu d'impressions sont collectées. Les résultats expérimentaux obtenus valident les idées développées dans notre approche.

Comme perspectives, nous entendons améliorer notre approche de différentes manières. Nous projetons ainsi de tester différents algorithmes de classification pour la détermination des zones compatibles. Nous projetons également d'utiliser des algorithmes d'alignement d'empreintes qui se basent sur d'autres caractéristiques que les minuties pour bénéficier à la fois des informations qu'apportent les minuties et des informations qu'apportent les autres caractéristiques. Finalement, nous entendons tester notre approche sur d'autres bancs de tests.

Remerciements Ces travaux de recherche sont effectués dans le cadre d'une thèse MOBIDOC financée par l'U.E. dans le cadre du programme PASRI.

Références

1. http :llwww.neurotechnologija.com/verifinger.html/.
2. Tai Pang Chen, Xudong Jiang, and Wei-Yun Yau. Fingerprint image quality analysis. In *Image Processing, 2004. ICIP'04. 2004 International Conference on*, volume 2, pages 1253–1256. IEEE, 2004.
3. Kyoungtaek Choi, Hee-seung Choi, and Jaihie Kim. Fingerprint mosaicking by rolling and sliding. In *International Conference on Audio-and Video-Based Biometric Person Authentication*, pages 260–269. Springer, 2005.
4. Kyoungtaek Choi, Heeseung Choi, Sangyoun Lee, and Jaihie Kim. Fingerprint image mosaicking by recursive ridge mapping. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(5) :1191–1203, 2007.

5. Marcos Faundez-Zanuy. Data fusion in biometrics. *IEEE Aerospace and Electronic Systems Magazine*, 20(1) :34–38, 2005.
6. Hartwig Fronthaler, Klaus Kollreider, and Joseph Bigun. Automatic image quality assessment with application in biometrics. In *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*, pages 30–30. IEEE, 2006.
7. Anil Jain and Arun Ross. Fingerprint mosaicking. In *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, volume 4, pages IV–4064. IEEE, 2002.
8. Xudong Jiang and Wee Ser. Online fingerprint template improvement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8) :1121–1126, 2002.
9. Yong Li, Jianping Yin, En Zhu, Chunfeng Hu, and Hui Chen. Score based biometric template selection and update. In *2008 Second International Conference on Future Generation Communication and Networking*, volume 3, pages 35–40. IEEE, 2008.
10. Dario Maio, Davide Maltoni, Raffaele Cappelli, James L. Wayman, and Anil K. Jain. Fvc2000 : Fingerprint verification competition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3) :402–412, 2002.
11. Davide Maltoni, Dario Maio, Anil Jain, and Salil Prabhakar. *Handbook of fingerprint recognition*. Springer Science & Business Media, 2009.
12. Yiu Sang Moon, Hoi-Wo Yeung, KC Chan, and SO Chan. Template synthesis and image mosaicking for fingerprint registration : an experimental study. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*, volume 5, pages V–409. IEEE, 2004.
13. Daniel Peralta, Mikel Galar, Isaac Triguero, Daniel Paternain, Salvador García, Edurne Barrenechea, José M Benítez, Humberto Bustince, and Francisco Herrera. A survey on fingerprint minutiae-based local matching for verification and identification : Taxonomy and experimental evaluation. *Information Sciences*, 315 :67–87, 2015.
14. Herbert Ramoser, B Wachmann, and Horst Bischof. Efficient alignment of fingerprint images. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 3, pages 748–751. IEEE, 2002.
15. Arun Ross, Samir Shah, and Jidnya Shah. Image versus feature mosaicing : A case study in fingerprints. In *Defense and Security Symposium*, pages 620208–620208. International Society for Optics and Photonics, 2006.
16. LinLin Shen, Alex Kot, and WaiMun Koo. Quality measures of fingerprint images. In *International Conference on Audio-and Video-Based Biometric Person Authentication*, pages 266–271. Springer, 2001.
17. Tamer Uz, George Bebis, Ali Erol, and Salil Prabhakar. Minutiae-based template synthesis and matching for fingerprint authentication. *Computer Vision and Image Understanding*, 113(9) :979–992, 2009.
18. Wei-Yun Yau, Kar-Ann Toh, Xudong Jiang, Tai-Pang Chen, and Juwei Lu. On fingerprint template synthesis. In *Proceedings of Sixth International Conference on Control, Automation, Robotics and Vision (ICARCV 2000). Singapore*, pages 5–8, 2000.
19. Yong-liang Zhang, Jie Yang, and Hong-tao Wu. A hybrid swipe fingerprint mosaicing scheme. In *International Conference on Audio-and Video-Based Biometric Person Authentication*, pages 131–140. Springer, 2005.