Machine Learning with Applications 7 (2022) 100210

Contents lists available at ScienceDirect

Machine Learning with Applications

journal homepage: www.elsevier.com/locate/mlwa

A-iLearn: An adaptive incremental learning model for spoof fingerprint detection

Shivang Agarwal [a,*], Ajita Rattani [b], C. Ravindranath Chowdary [a,*]

[a] Department of Computer Science and Engineering, Indian Institute of Technology (BHU), Varanasi, 221005, India
[b] Department of Electrical Engineering and Computer Science, Wichita State University, USA

ARTICLE INFO

Keywords:
Incremental learning
Stability-plasticity dilemma
Catastrophic forgetting
Spoof fingerprint detection

ABSTRACT

Incremental learning enables the learner to accommodate new knowledge without retraining the existing model. It is a challenging task that requires learning from new data and preserving the knowledge extracted from the previously accessed data. This challenge is known as the stability-plasticity dilemma. We propose A-iLearn, a generic model for incremental learning which overcomes the stability-plasticity dilemma by carefully integrating the ensemble of base classifiers trained on new data with the current ensemble without retraining the model from scratch using entire data. We demonstrate the efficacy of the proposed A-iLearn model on spoof fingerprint detection application. One of the significant challenges associated with spoof fingerprint detection is the performance drop on spoofs generated using new fabrication materials. A-iLearn is an adaptive incremental learning model that adapts to the features of the "live" and "spoof" fingerprint images and efficiently recognizes the new spoof fingerprints and the known spoof fingerprints when the new data is available. To the best of our knowledge, A-iLearn is the first attempt in incremental learning algorithms that adapts to the properties of data for generating a diverse ensemble of base classifiers. From the experiments conducted on standard high-dimensional datasets LivDet 2011, LivDet 2013 and LivDet 2015, we show that the performance gain on new fake materials is significantly high. On average, we achieve 49.57% improvement in accuracy between the consecutive learning phases.

1. Introduction

Incremental learning is a process of learning in the presence of new data while retaining the knowledge acquired from previously seen data. Incremental learning is useful for applications that require accessing a huge amount of data in regular chunks because it does not need to retrain the model on the entire data when the model needs to expand progressively. For instance, in spoof fingerprint detection, where the task is to classify the fingerprint images as "live" and "spoof", the learning model is expected to learn incrementally from fingerprint images generated using novel fabrication materials. Therefore, the learning model must preserve the knowledge H_t extracted from previously seen data D_{Train_t} of live and spoof fingerprint images while learning from the upcoming data $D_{Train_{t+1}}$ without accessing D_{Train_t}. After learning from $D_{Train_{t+1}}$, the knowledge H_{t+1} must be carefully integrated with H_t. Therefore,

$$H_{t+1} \leftarrow D_{Train_{t+1}} \cup H_t \tag{1}$$

Polikar, Upda, Upda, and Honavar (2001) define a set of properties that anefficient incremental learning algorithm must possess: (1) it must learn new knowledge from upcoming data, (2) it should not require to access old training data, (3) it should preserve the previously acquired knowledge, and (4) it should be able to accommodate new concepts that may be available in the novel data. To address these properties, we propose A-iLearn, an adaptive incremental learning algorithm based on ensemble learning, which learns from the new data while retaining the previous knowledge without requiring to access the old data.

The major challenge in incremental learning is to overcome the stability–plasticity dilemma, where stability signifies retaining the previously acquired knowledge, and plasticity signifies learning from new data (Polikar et al., 2001). Therefore, an ideal approach for incremental learning must find a balance between stability and plasticity. Focusing only on plasticity may lead to a situation called *catastrophic forgetting* where the learning model forgets the previously acquired knowledge while learning from new data (Win, Li, Chen, Viger, & Li, 2020). At the same time, concentrating only on stability may lead to the inability of capturing comprehensive knowledge from the latest data (Li, Huang, Xiong, Ren, & Zhu, 2020).

Another challenge in incremental learning is learning in the presence of concept drift (Elwell & Polikar, 2011; Iwashita, de Albuquerque, & Papa, 2019; Yuan, Wang, Zhuang, Zhu, & Hao, 2018). Concept drift

[*] Corresponding authors.
E-mail addresses: shivanga.rs.cse16@iitbhu.ac.in (S. Agarwal), ajita.rattani@wichita.edu (A. Rattani), rchowdray.cse@iitbhu.ac.in (C.R. Chowdary).

https://doi.org/10.1016/j.mlwa.2021.100210
Received 20 January 2021; Received in revised form 12 October 2021; Accepted 4 November 2021
Available online 17 November 2021

is a situation where the underlying data distribution changes over time, such that

$$p_{t+1}(x, w) \neq p_t(x, w) \tag{2}$$

where, x represents an instance, w is the class label associated with x, and t is the timestamp.

Incremental learning can be applied in various applications such as credit card fraud detection (Chan & Stolfo, 1998; Pozzolo, Caelen, Borgne, Waterschoot, & Bontempi, 2014), object recognition in image processing (He & Chen, 2008), video surveillance in computer vision (Lu, Boukharouba, Boonært, Fleury, & Lecœuche, 2014), interactive kinaesthetic teaching in robotics (Saveriano, An, & Lee, 2015), automated annotation for video and speech tagging (Gepperth & Hammer, 2016), science article recommendation (Luo, Xia, & Zhu, 2012), image recognition (Roy, Panda, & Roy, 2020), text classification (Shan, Xu, Yang, Jia, & Xiang, 2020), object learning (Stojanov et al., 2019), social network analysis (Peng et al., 2018), crypto-ransomware detection (Al-rimy, Maarof, & Shaid, 2019) and similar applications. It is particularly useful in applications where it is required to learn from data in multiple phases, one chunk at a time or the applications where the size of data is so vast that it requires to be broken into numerous parts and accessed in multiple phases. In addition, incremental learning is also useful in applications where the old data is no longer available. Therefore the model cannot be retrained from the entire data; instead, it must be incrementally updated.

The learning behaviour varies depending on the application to which it has been applied. Learning methods can be grouped into three categories based on applications with different data requirements:

1. Applications that require access to the previously seen data. Predicting the stock exchange is one of these applications where we need yesterday's data and today's data to predict for tomorrow (Chen & Hao, 2017).
2. Applications that need access only to the knowledge extracted from the previously accessed data but not the actual data itself. Cancer diagnosis belongs to such an application where we need only the knowledge extracted from the previous bunch of data along with the current data (Gu, Quan, Gu, Sheng, & Zheng, 2018).
3. Applications that require discarding the previous data and building a new model strictly based on the current data. Analysing the current trends on social media such as Twitter is one of these applications where the previously accessed data become insignificant over time, and the learning model must be built entirely on the current data (Saha & Sindhwani, 2012).

The difference in the nature of incremental learning and conventional single-phase learning can be understood by the above classification. A model following the incremental learning paradigm requires learning from multiple chunks of data, extracting the knowledge from them, discarding the previously seen data, and using the already acquired knowledge and the current data. Another variant of learning paradigms is online learning which learns from every incoming instance. Online learning is a subarea of incremental learning that is useful in situations where limited data is provided and shorter runtime is expected from the model (He, Mao, Shao, & Zhu, 2020). Online learning is useful in situations where streaming data arrives at high-velocity (Nallaperuma et al., 2019). On contrary to the chunk-based incremental learning, online learning operates by repetitively drawing a random training example to tune the learning parameters. Although online algorithms can process a large amount of instances in shorter time but they are usually incapable of fully optimizing the cost function based on observed instances (Bottou & LeCun, 2003).

On the other hand, conventional single-phase learning requires discarding the existing classifier whenever new data is available and retraining the model from the ground truth.

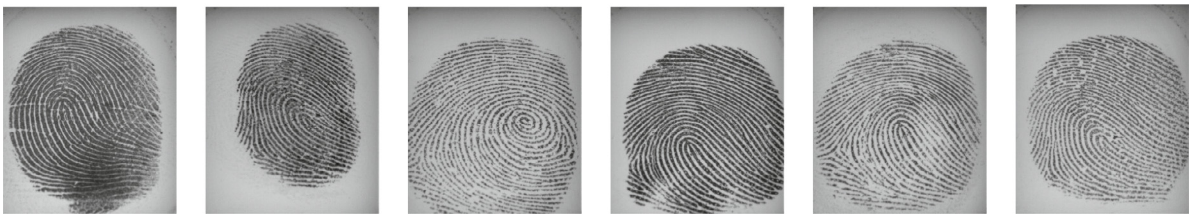
Transfer learning provides a solution for transferring the knowledge acquired on a training data with a feature space to use it on new data with different feature space (Pan & Yang, 2010). In addition, unlike incremental learning, transfer learning usually deals with situations where the source and target data distributions are different (Zhuang et al., 2021). Domain adaptation is considered to be a sub-category of transfer learning. It is particularly useful in situations where the source and target feature spaces are the same, but the data distributions are different (Pan, Tsang, Kwok, & Yang, 2011).

Presentation attack detection is an emerging research domain, which involves fingerprint or face spoof detection where spoofs of new fabrication materials are frequently observed by the learning models (Agarwal & Chowdary, 2020; Jia, Guo, Xu, & Wang, 2020; Sharma & Dey, 2019; Yu, Yao, Pei, & Jia, 2019). Therefore, we position this study to fall under category 2 of the above classification. Existing studies suggest that fingerprint recognition systems are vulnerable to attacks by spoof fingerprints made of different materials such as gelatin, silicone, latex etc. Jia et al. (2014) and Menotti et al. (2015). As represented by Fig. 1, it is not possible for us to manually distinguish between live and spoof fingerprints generated using these materials. Further, the performance of the spoof fingerprint detector substantially degrades on the novel spoof materials (Kho, Lee, Choi, & Kim, 2019; Rattani, Scheirer, & Ross, 2015). Incremental learning is one of the solutions to mitigate performance degradation due to evolving spoofing techniques by using novel fabrication materials. Therefore, the spoof fingerprint detection application has been chosen for demonstrating the efficacy of incremental learning.

Our adaptive incremental learning model A-iLearn yields a robust spoof detector that can identify spoof fingerprints generated from fabrication materials unknown to the current model in spoof fingerprint detection. We exploit the incremental learning scenario by considering two learning phases. The model learns images from live category and spoof images of two fabrication materials in the first phase. In the later phase, the model learns the spoof images of the remaining fabrication materials. The idea is to test the ability of the model to maintain stability and plasticity while obtaining performance gain on novel spoof materials in multiple phases. In an ideal scenario, the incremental learner must not observe a significant performance degradation on the known data while improving its performance on new data in subsequent learning phases (Wei, Liu, Xiang, Duan, Zhao et al., 2020).

The contributions made by this study are as follows:

1. We propose a novel incremental learning model A-iLearn, which is adaptive towards the similarity inherently present in the data and is capable of overcoming the classic stability–plasticity dilemma. The proposed model produces lower performance degradation and higher performance improvement while learning in multiple phases.
   - We perform clustering on the training data to generate clusters of instances and use these clusters to train RBF SVMs as base classifiers which results in an ensemble of diverse classifiers. In our observations, these base classifiers are free from catastrophic forgetting, i.e. while learning from new data, there is no significant performance loss concerning the previously trained instances.
2. Our proposed A-iLearn does not need to retrain the model from scratch while introducing new data to it. As we use an ensemble of base classifiers, it offers us a high degree of reliability and robustness. The new knowledge is added by carefully integrating another ensemble into the model.
3. A-iLearn for spoof fingerprint detection does not need to access the previously seen fingerprints while learning the new fingerprints, which results in low memory requirements. In addition, it discards the poorly performing base classifiers and uses only the relevant ones to save the storage and improve the classification accuracy.



(a) Live　(b) Ecoflex　(c) Gelatin　(d) Latex　(e) Silgum　(f) WoodGlue

Fig. 1. Visual comparison between live and spoofs created using various spoof materials.

2. Related work

2.1. Incremental learning

Incremental learning has benefited many applications that require learning in pieces. In Chowdary and Kumar (2008), the authors propose a method for an incremental summary generation where the task is to update the current summary of documents after encountering a new document. Their proposed approach finds the most significant sentence in the document that can be replaced with a sentence in the current summary making it more accurate and updated.

Most of the approaches proposed for incremental learning so far make use of ensemble learning, where a set of classifiers are learned from each chunk of training data. Learn++ (Polikar et al., 2001) is a family of algorithms widely used for various applications involving incremental learning (Elwell & Polikar, 2011; Muhlbaier, Topalis, & Polikar, 2009; Polikar et al., 2001). Learn++ algorithms address the issues like the stability–plasticity dilemma, catastrophic forgetting, concept drift etc. Learn++ was initially proposed for training neural network pattern classifiers incrementally. It generates weak hypotheses and combines them using a weighted majority voting scheme. Learn++.NSE (Elwell & Polikar, 2011) enhances the previous version by accommodating concept drift in non-stationary environments. Learn++.NSE-SMOTE (Ditzler & Polikar, 2013) enhances the previous versions by overcoming the problem of class imbalance in the streaming data. Irrespective of its popularity and broad applicability, the Learn++ family of incremental learning models do not consider data properties while generating the base learners.

There are some cost-sensitive learning approaches proposed in the past that assign unequal misclassification costs to different classes (Gu, Sheng, & Li, 2015). Incremental learning becomes useful as it keeps track of the concept drift and manages it not to increase false negatives. Another approach for detecting concept drift for incremental learning on non-stationary environments is proposed in Yang, Al-Dahidi, Baraldi, Zio, and Montelatici (2020). The authors claim that the early detection of concept drift may result in improved accuracy.

Recently, there has been some research on handling the class imbalance while learning incrementally (hui Hou, kang Wang, yu Zhang, qiang Wang, & Li, 2020; Tharwat & Schenck, 2020). In Wu et al. (2019), the proposed learning model scales up to a large number of classes while managing the data imbalance between previously observed and new classes. While learning from the newly added data, it is essential to update the hypothesis accordingly. In Zhang et al. (2020), the old and new learning models are consolidated via a double distillation training objective. An unlabelled auxiliary data is exploited to consolidate the two models. In Li et al. (2020), the authors propose a dynamically updated ensemble algorithm for dealing with class imbalance and concept drift.

A-iLearn performs clustering to generate an ensemble of base classifiers in each learning phase. We claim that the advantages of clustering are two-fold: we can have a disjoint ensemble of classifiers (a must-have property in ensemble learning), and the base classifiers grasp the features of the data, which is useful while classifying similar but unknown test instances. The intuition is motivated by the cluster-class assumption (Chapelle, Schölkopf, & Zien, 2006; Rigollet, 2007) which states that “similar instances should share the same class label”.

Algorithm 1: Learning Incrementally using A-iLearn Algorithm

```
Data: training data $D_{Train} =< x_s, y_s >$,
      test data $D_{Test} =< x_t, y_t >$,
      validation data $D_{Valid} =< x_u, y_u >$,
      a clustering algorithm $C$,
      a classification algorithm $K$,
      number of base classifiers $n$,
      number of learning phases $p$.
Result: ensemble classifier $Z_f$
partition $D_{Train}$ and $D_{Test}$ into $p$ parts each;
$Z_0$= NULL;
for $i$= 1 to $p$ do
    $\{c_1, c_2, .., c_n\} \leftarrow C(D_{Train_i})$;
    for $j$=1 to $n$ do
        $k_j \leftarrow K(c_j)$;
        Check the accuracy $a_j$ of $k_j$ on $D_{Valid}$;
    $Z_i \leftarrow \{k_1, k_2, .., k_n\}_i$;
    $Z_i \leftarrow Z_i + Z_{i-1}$;
$Z_f \leftarrow Z_i$;
```

2.2. Spoof fingerprint detection

Most of the research in spoof fingerprint detection does not consider it an application of incremental learning where fingerprints with novel spoof materials are added to the model. Various single-phase spoof detectors have been proposed in the past (Gragnaniello, Poggi, Sansone, & Verdoliva, 2015; Jin, Li, Kim, & Park, 2011; Marasco & Ross, 2014; Nogueira, de Alencar Lotufo, & Campos Machado, 2016). The performance of single-phase spoof detectors degrades drastically when new fingerprints generated from novel spoof materials are introduced. We claim that the incremental aspect of fingerprint liveness detection must be studied so that the spoof detectors can accommodate new spoof materials efficiently. The recent research on spoof fingerprint detection shows the incremental behaviour by using the existing learning algorithms based on ensemble learning such as Learn++.NC (Kho et al., 2019; Muhlbaier et al., 2009), whereas some of the researchers find a way to modify SVMs to accommodate new knowledge to the model (Rattani et al., 2015). We claim that while learning the base classifiers of the ensemble, the properties intrinsic to the dataset must be considered to form clusters of instances. Therefore, we propose a new learning algorithm and use it for spoof fingerprint detection.

To the best of our knowledge, A-iLearn is the first incremental learning algorithm which considers adaptiveness towards the similarity present in the data to generate a diverse set of base classifiers and integrate the ensembles incrementally in subsequent learning phases.

3. A-iLearn: A generic model for incremental learning

The generic model of A-iLearn for incremental learning is described in Algorithm 1. This model is application-independent and can be applied to various applications. This paper considers an application where we fix the number of base classifiers to be generated in every learning phase p_i. Therefore we consider the number of base classifiers n as an input to the algorithm. Also, we take as inputs a clustering algorithm C and a classification algorithm K. The proposed algorithm is independent of the choice of clustering and classification algorithms.

To show the incremental behaviour of A-iLearn, we need to partition the original training dataset D_{Train} into p parts to be accessed in p learning phases (step 1 in Algorithm 1). We partition the test dataset D_{Test} as well accordingly. The target is to demonstrate the incremental behaviour by adding a new batch of training data in each learning phase and testing the performance of the learned model on $D_{Test_{i+1}}$ as well as on D_{Test_i}.

Later, we perform clustering on each of the partitioned training set D_{Train_i} that yields a set of clusters $c_1, c_2, .., c_n$. Next, we train base classifiers on each generated cluster c_j by using a classification algorithm K (steps 4–6 of the algorithm). Therefore, we generate an ensemble of base classifiers where the diversity among the base classifiers is high due to clustering. As pointed out by Polikar et al. (2001), if each classifier is trained on a different subset of training data based on some distribution, then it is highly probable that a misclassified instance is classified correctly by another classifier. Therefore, it is always advantageous to have a diverse set of base classifiers.

After learning in every phase, we first test the accuracy of the ensemble of base classifiers generated in the current phase on validation data held out from training (step 7 of the algorithm). The purpose of using validation data is to assign weights to the base classifiers based on their performance. We use Eqs. (3) and (4) to assign weights to the classifiers.

$$w_y^x = \sum_{i=1}^{n} a_{iy}^x \tag{3}$$

where, w_y^x is the total weight associated with the class label y for an instance x. n is the number of base classifiers and a_{iy}^x is the accuracy of ith base classifier which has predicted the class label y for instance x on validation data. The final class label y_f^x determined by the weighted majority is given by Eq. (3):

$$y_f^x = argmax_y(w_y^x) \tag{4}$$

We test the accuracy of the ensemble Z_i generated in ith learning phase (step 8) by classifying the instances of test data using Eqs. (3) and (4). Based on a threshold on the performance, the poorly performing classifiers are discarded so that they do not participate in the voting process. Similarly, in the next phase, we generate another ensemble of base classifiers Z_{i+1}, trained on the newly added data $D_{Train_{i+1}}$. We test the performance of individual base classifiers on validation data and merge the qualifying base classifiers with the existing ensemble from the previous phase (step 9). All the qualifying classifiers are tested on the test data to demonstrate performance improvement. The idea is to highlight that as the model proceeds to learn $D_{Train_{i+1}}$, its performance is increased as tested on the new data without considerable performance loss on D_{Train_i}.

4. A-iLearn for spoof fingerprint detection

The incremental ability of the proposed algorithm is achieved by considering multiple learning phases. The performance degradation for the previously known knowledge and the performance improvement for the newly learned data are observed while moving to subsequent learning phases.

The schematic diagram explaining the working mechanism of A-iLearn on spoof fingerprint detection is given in Fig. 2. In this study, we have used LivDet 2011 (Yambay, Ghiani, Denti, Marcialis, Roli et al., 2012), LivDet 2013 (Ghiani et al., 2013), and LivDet 2015 (Mura, Ghiani, Marcialis, Roli, Yambay et al., 2015) datasets. We partition each training data into two parts: Known Fake (KF) and New Fake (NF). KF consists of 1000 live instances and 400 spoof instances belonging to two of the spoof subcategories. NF consists of 600 spoof instances of the remaining subcategories. Test data is partitioned in the same manner (i.e., $Test_1$: 1000 Live + 400 Spoof, $Test_2$: 600 Spoof).

In the first phase, when the model is trained on KF, we test its accuracy on both $Test_1$ and $Test_2$. The idea is to see the difference in both accuracies as in the first phase, the model is trained only on KF; it must yield good accuracy on it but poor accuracy on NF. In the second phase, when new fake data is introduced to the training model, we need to accommodate it by managing two challenges: first, the knowledge acquired in the first phase must not be lost. Therefore, the accuracy of the updated model must not decrease drastically when tested on KF test data. This shows that the model possesses better stability. Second, the model must be able to learn from the new fake data added to the training model. Therefore the accuracy on NF must increase significantly, which shows that the model possesses better plasticity. The components of the proposed framework are explained as follows:

4.1. Feature extraction

1. Local Binary Patterns (LBP) are a local texture descriptor that is widely used for fingerprint liveness detection (Ahonen, Hadid, & Pietikainen, 2006; Nogueira et al., 2016). LBP is an illumination invariant descriptor that determines an image's texture by labelling each pixel with a binary value based on the thresholds on the neighbouring pixels. It considers the central pixel as the threshold, and based on that, it assigns the binary values to the neighbouring pixels. LBP value of the pixel is calculated by adding up the element-wise multiplications of the binary values with their weights.
2. Local Phase Quantization (LPQ) can be an effective method for detecting the liveness of a fingerprint image as it is insensitive to the blurring effect (Ghiani, Marcialis, & Roli, 2012; Ojansivu & Heikkilä, 2008). LPQ features consider the spectral differences between live and spoof fingerprint images.
3. Binarized Statistical Image Features (BSIF) is a method of constructing local image features to encode the texture information from the images (Kannala & Rahtu, 2012). The descriptors are determined by the statistical properties of natural image patches. For a particular image, BSIF computes a binary string for the pixels and use it as a local descriptor of the image intensity pattern in the pixel's surroundings.
4. ResNet-50 (He, Zhang, Ren, & Sun, 2016) is a deep Residual Network originally designed for object recognition. ResNet-50 has been pretrained on ImageNet database (Russakovsky et al., 2015). By extracting the features using ResNet-50, we utilize transfer learning for spoof fingerprint detection. ResNet architecture is used for deep feature extraction as it is among the efficient Convolutional Neural Networks introduced till now. ResNet utilizes skip connections or shortcuts to jump over layers to avoid the problem of vanishing gradients.

For a detailed comparison of handcrafted features v/s deep features for fingerprint liveness detection, we encourage the readers to refer Agarwal, Rattani, and Chowdary (2021). The current manuscript is different from Agarwal et al. (2021), as the aim is to propose a novel A-iLearn algorithm which can accommodate new knowledge without retraining the existing model. We demonstrated the efficacy of A-iLearn for fingerprint liveness detection using both hand-crafted and deep features.

4.2. Ensemble generation

After feature extraction, the proposed incremental learning algorithm generates the ensemble of base classifiers. The ensemble is created by using the following components:

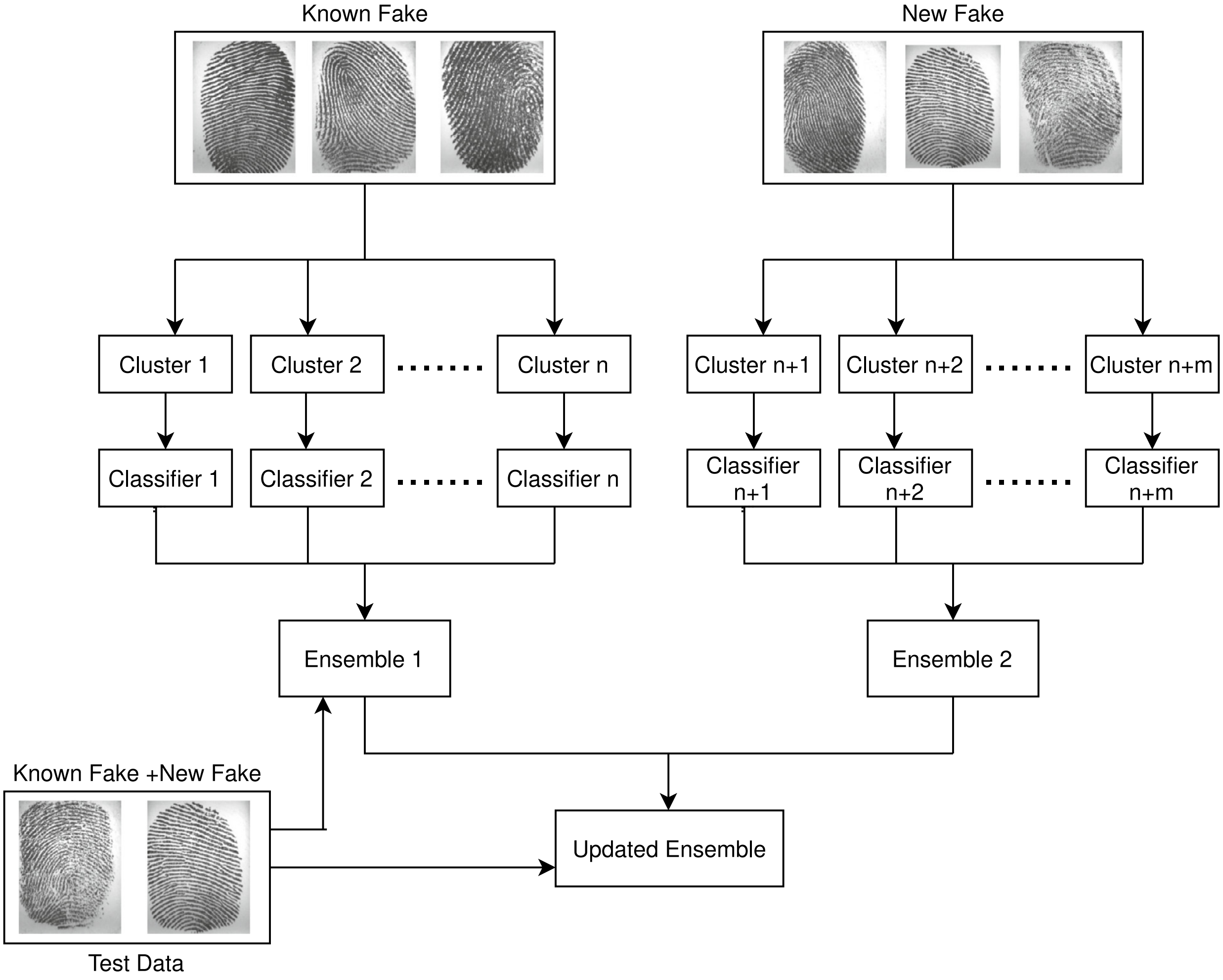


Fig. 2. Schema of the proposed A-iLearn incremental learning algorithm for spoof fingerprint detection.

- Training data $D_{Train} = (x_1, y_1), \dots .(x_n, y_n)$ contains training examples belonging to both "live" and "spoof" classes, where x_i is a set of attributes generated from an image feature extraction algorithm and y_i is the corresponding class label.
- A clustering algorithm C is used to cluster the training examples based on the similarity of records in the data. The target is to create a group of clusters $c_1, .., c_n$ where the examples belonging to one cluster possess similar values of attributes, whereas the examples belonging to different clusters possess different values of the attributes defined by image features. We strongly recommend using a clustering algorithm that does not require defining the number of clusters n apriori so that the clusters are naturally formed based on the similarities, but depending upon the application, the number of base clusters may be known apriori.
- A classification algorithm K to train the base classifiers on each c_n. K uses each c_n to generate a base classifier which is used to make a decision individually. In this way, the decision boundary of each base classifier is different from others, resulting in an ensemble of diverse base classifiers (Melville & Mooney, 2005). Later these decisions are integrated by using weighted majority voting to decide for the whole ensemble.

5. Experimental results and discussion

5.1. Dataset

The description of the datasets used in this paper is given in Table 1. We use LivDet datasets of three years: LivDet 2011 (Yambay et al., 2012), LivDet 2013 (Ghiani et al., 2013) and LivDet 2015 (Mura et al., 2015), which were used in fingerprint liveness detection competition conducted in consecutive years. This competition aims to compare software-based fingerprint liveness detection methodologies and fingerprint systems that incorporate liveness detection capabilities. The datasets consist of fingerprint images broadly classified into two classes: "live" and "spoof". These fingerprints are tested on various biometric sensors: Biometrika, DigitalPersona, ItalData, Sagem etc. For all the sensors, we have approximately 1000 fingerprint images each for both of the classes in training and testing.

Further, the images belonging to the spoof class can be categorized in multiple sub-categories based on the materials used for creating the spoof or fake fingerprint. These materials are gelatin, latex, playdoh, wood glue, silicone etc. In LivDet 2011 (Yambay et al., 2012) and LivDet 2013 (Ghiani et al., 2013), each dataset has 200 images belonging to each of the five sub-categories, whereas in LivDet 2015 (Mura et al., 2015), each dataset has 250 images belonging to each of the four sub-categories.

5.2. Experimental settings

As mentioned in Section 5.1, we partition each of the training and testing data belonging to a particular sensor (e.g. Biometrika, DigitalPersona, etc.) into two parts: I. Known Fake (KF) and II. New Fake (NF). In LivDet 2011 (Yambay et al., 2012) and LivDet 2013 (Ghiani et al., 2013), we have five sub-categories in the spoof class; therefore, we have ten possible combinations of training datasets for the known

Table 1
The training and testing protocol of the experiments.

Database		Live (Train/Test)	Spoof (Train/Test)
LivDet2011 (Yambay et al., 2012)	Biometrika	1000/1000	1000/1000 (ecoflex, gelatin, latex, silgum, wood glue)
	DigitalPersona	1000/1000	1000/1000 (gelatin, latex, playdoh, silicone, wood glue)
	ItalData	1000/1000	1000/1000 (ecoflex, gelatin, latex, silgum, wood glue)
	Sagem	1000/1000	1000/1000 (gelatin, latex, playdoh, silicone, wood glue)
LivDet2013 (Ghiani et al., 2013)	Biometrika	1000/1000	1000/1000 (ecoflex, gelatin, latex, modasil, wood glue)
	ItalData	1000/1000	1000/1000 (ecoflex, gelatin, latex, modasil, wood glue)
LivDet2015 (Mura et al., 2015)	Biometrika	1000/1000	1000/1000 (ecoflex, gelatin, latex, wood glue)
	DigitalPersona	1000/1000	1000/1000 (ecoflex, gelatin, latex, wood glue)

Table 2
Partitioning of the datasets in Phase I and Phase II for evaluation of the A-iLearn algorithm.

Sr. No.	LivDet2011 (Yambay et al., 2012), LivDet2013 (Ghiani et al., 2013)		LivDet2011 (Yambay et al., 2012)		LivDet2015 (Mura et al., 2015)	
	Biometrika, ItalData		Digital, Sagem		Biometrika, Digital	
	Phase I	Phase II	Phase I	Phase II	Phase I	Phase II
1	Live+Ecoflex+Gelatin	Latex+Silgum+Woodglue	Live+Gelatin+Latex	Playdoh+Silicone+Woodglue	Live+Ecoflex+Gelatin	Latex+Woodglue
2	Live+Ecoflex+Latex	Gelatin+Silgum+Woodglue	Live+Gelatin+Playdoh	Latex+Silicone+Woodglue	Live+Ecoflex+Latex	Gelatin+Woodglue
3	Live+Ecoflex+Silgum*	Gelatin+Latex+Woodglue	Live+Gelatin+Silicone	Latex+Playdoh+Woodglue	Live+Ecoflex+Woodglue	Gelatin+Latex
4	Live+Ecoflex+Woodglue	Gelatin+Latex+Silgum	Live+Gelatin+Woodglue	Latex+Playdoh+Silicone	Live+Gelatin+Latex	Ecoflex+Woodglue
5	Live+Gelatin+Latex	Ecoflex+Silgum+Woodglue	Live+Latex+Playdoh	Gelatin+Silicone+Woodglue	Live+Gelatin+Woodglue	Ecoflex+Latex
6	Live+Gelatin+Silgum	Ecoflex+Latex+Woodglue	Live+Latex+Silicone	Gelatin+Playdoh+Woodglue	Live+Latex+Woodglue	Ecoflex+Gelatin
7	Live+Gelatin+Woodglue	Ecoflex+Latex+Silgum	Live+Latex+Woodglue	Gelatin+Playdoh+Silicone		
8	Live+Latex+Silgum	Ecoflex+Gelatin+Woodglue	Live+Playdoh+Silicone	Gelatin+Latex+Woodglue		
9	Live+Latex+Woodglue	Ecoflex+Gelatin+Silgum	Live+Playdoh+Woodglue	Gelatin+Latex+Silicone		
10	Live+Silgum+Woodglue	Ecoflex+Gelatin+Latex	Live+Silicone+Woodglue	Gelatin+Latex+Playdoh		

fake. In LivDet 2015 (Mura et al., 2015), there are four sub-categories of spoof class; therefore we have six such combinations. In the first experimental setting, we partition the training data into two parts and learn each of them in two learning phases (e.g., I. Live + Ecoflex + Gelatin, II. Latex + Silicone + Woodglue). The description of the partitioning of these datasets is given in Table 2. In every phase, the trained model is tested on two test datasets created using the same setup. In the second experimental setting, we retrain the model in the second learning phase using the entire training data. We use the second experimental setting as the benchmark for the proposed model. Therefore, the motivation is to have competitive performance without the need for retraining the model using the entire data.

We use ResNet-50 (He et al., 2016) for extracting deep features from the fingerprint images. LBP, LPQ and BSIF features were extracted using MATLAB. The extracted features are converted into arff files to make them WEKA compatible. We use Waikato Environment for Knowledge Analysis (Weka) (Hall, Frank, Holmes, Pfahringer, Reutemann et al., 2009) to classify the images into "live" and "spoof" classes.

The proposed A-iLearn is not restricted to any particular clustering or classification algorithm. In our experiments, we use SimpleK-Means (Arthur & Vassilvitskii, 2007) clustering algorithm with k=2. We use SMO (John Platt's sequential minimal optimization algorithm for training a support vector classifier) (Platt, 1998) as the classification algorithm. SVMs have been an appropriate choice for classifying fingerprint images as live and spoof (Agarwal & Chowdary, 2020, 2021; Kho et al., 2019; Rattani et al., 2015). For every dataset, we report the overall accuracy of the model on known fake (KF) and new fake (NF) data as well as the bonafide presentation classification error rate (BPCER) and attack presentation classification error rate (APCER), where bonafide presentation = "live", and attack presentation = "spoof".

The experiments are conducted on a Linux machine with 26 GB memory.

5.3. Results

In this section, we provide the experimental results conducted on three high dimensional datasets. To demonstrate the incremental behaviour of the proposed model, we learn the data in two phases. In the first phase, the model is trained over instances of the "Live" category and instances belonging to two sub-categories of the "Spoof" class. In the first phase, when the learned model is tested using the known sub-categories of spoof class, we call it "I. Known Fake". We also test the learned model on the remaining sub-categories on which the model is not yet trained; we call it "I. New Fake". In the second phase, we train the model on the remaining sub-categories of spoof fingerprints and integrate the hypotheses with the existing model. Again we test this updated model's performance on the same test sets; we call them "II. Known Fake" and "II. New Fake". As A-iLearn does not require to access the whole training data in the second phase, the count of "Live" instances in the second training data is 0; therefore, in place of BPCER in New Fake, we have written N/A.

Tables 3 and 4 describe the experimental results for A-iLearn on various datasets of LivDet database while using LBP, LPQ, BSIF and ResNet-50 features.[1]

Table 3 represents the average of A-iLearn's stability–plasticity values for all combinations of LivDet2011 (Yambay et al., 2012) datasets as described in Table 2. In Table 3, we report the average performance of all features for a particular sensor (e.g., Biometrika, DigitalPersona, etc.). The feature-level comparison for individual sensors is given in Fig. 3. As a baseline for comparison, we report the learning model's performance while retraining it using the entire data (e.g., Bio-RT). As we retrain the model in the second phase of learning, the performance values for the first phase are the same as without retraining.

While using LBP features on Biometrika, the average performance degradation for Known Fake (KF) from the first phase to the second phase is 3.34%, whereas the performance improvement for New Fake (NF) is 29.75%. On LPQ features, the average performance degradation for KF from the first phase to the second phase is 5.57%, whereas the performance improvement for NF is 28.03%. A-iLearn performs reasonably well on **LivDet2011** (Yambay et al., 2012) Biometrika dataset using ResNet-50 features, with which it yields 57.23% performance

[1] Note that Sagem dataset has 1036 spoof images in the test set and DigitalPersona dataset has 1004 live images in the train set. For LBP, LPQ, and BSIF features, we have considered the original quantity of images, but for ResNet, we have considered 1000 images from each category.

improvement from the first phase to the second phase. Also, it is evident that on the Biometrika dataset, the best local feature is BSIF, which yields only 1.86% performance degradation on KF whereas 32.69% performance improvement on NF.

On the DigitalPersona dataset, while using LBP features, the performance of the model on KF is increased by 8.97% in the second phase, the performance on NF is improved by 15.8% in the second phase. The results of the LPQ feature are significantly well on KF. There is no performance degradation while moving to the second phase; rather, the performance is improved by 2.65%. Also, the performance is improved by 39.86% on NF in the second phase. The results on the BSIF feature are adequate as the performance degradation on KF is only 7.05%, and the performance improvement is 34.83% with decent overall accuracy. While using ResNet-50 features on the DigitalPersona dataset, the performance is improved on KF by 3.19%, whereas on NF in the second phase, it is improved by 32.31%.

On the ItalData dataset, using LBP features, the model yields satisfying stability and reasonable plasticity, but the accuracy on NF is not adequate. Using LPQ features, the performance degradation on KF is only 2.59%, and the performance improvement on NF is 6.55%. Using BSIF features, we get outstanding results with only 3.71% performance degradation on KF but 56.37% performance improvement on NF. While using ResNet-50 features on ItalData 2011 dataset, the performance drop on KF from the first phase to the second phase is 4.99%, whereas the performance improvement on NF is 2.49%.

On the Sagem dataset, while using LBP features, we get slightly higher performance degradation (13.71%) on KF, which affects the stability of the model, but the plasticity is significantly well with a 49.88% performance improvement on NF. While using LPQ features, we get 7.71% performance improvement on KF with 92.67% average accuracy and 47.30% improvement on NF, resulting in sound plasticity. The best results we achieve while using BSIF features. We get a 3.84% improvement on KF and 50.45% improvement on NF, resulting in high plasticity with no compromise on stability. While using ResNet-50 features, the performance of A-iLearn on KF increases by 5.88% in the second phase, and on NF the performance is increased by 45.99%.

We test the performance of A-iLearn on some samples of **LivDet2013** (Ghiani et al., 2013) and **LivDet2015** (Mura et al., 2015) datasets. On the LivDet2013 Biometrika dataset, using BSIF features, we get a performance improvement of 2.87% on NF with adequate overall performance. Using LPQ features, we get an improvement of 11.95% on NF in the second phase. Using ResNet-50 features on the LivDet2013 Biometrika dataset, we improve by 187.68% on NF with 99.02% average performance on NF in the second phase.

On the LivDet2013 ItalData dataset, while using the BSIF feature, the performance improvement is not significant, but the overall performance is reasonably well. While using LPQ features, we get an improvement of 2.45% with excellent overall performance. While using ResNet-50 features, we get a 34.38% increase in the performance on NF in the second phase.

On the LivDet2015 Biometrika dataset, using BSIF features, we get an increase of 4.47% on NF in the second phase with excellent overall performance. Using LPQ features, the performance is increased by 34.11% in the second phase. While using ResNet features, the performance is increased by 35.56% on NF in the second phase. On the LivDet2015 DigitalPersona dataset, using BSIF features, there is no significant improvement, but the performance is 90.4% in the second phase. Using LPQ features, there is an increase of 13.07% on NF in the second phase. Using ResNet-50 features, we get an increase of 4.25% on NF in the second phase.

5.4. Feature-level comparison

Fig. 3 represents the comparison among various features used with A-iLearn on LivDet 2011 (Yambay et al., 2012) datasets. We emphasize the percentage gain on NF and percentage loss on KF in subsequent phases. Ideally, the features on which the percentage gain is high and percentage loss is low are the most suited for the application. From Fig. 3, we can see on Biometrika, using ResNet features, we get a performance gain of 57.23%, whereas the loss in performance on KF is −3.29%. The '−' symbol represents that instead of performance loss on KF, we observe a gain of −3.29%. Among the handcrafted features, LBP yields the highest percentage gain and BSIF yields the lowest percentage loss.

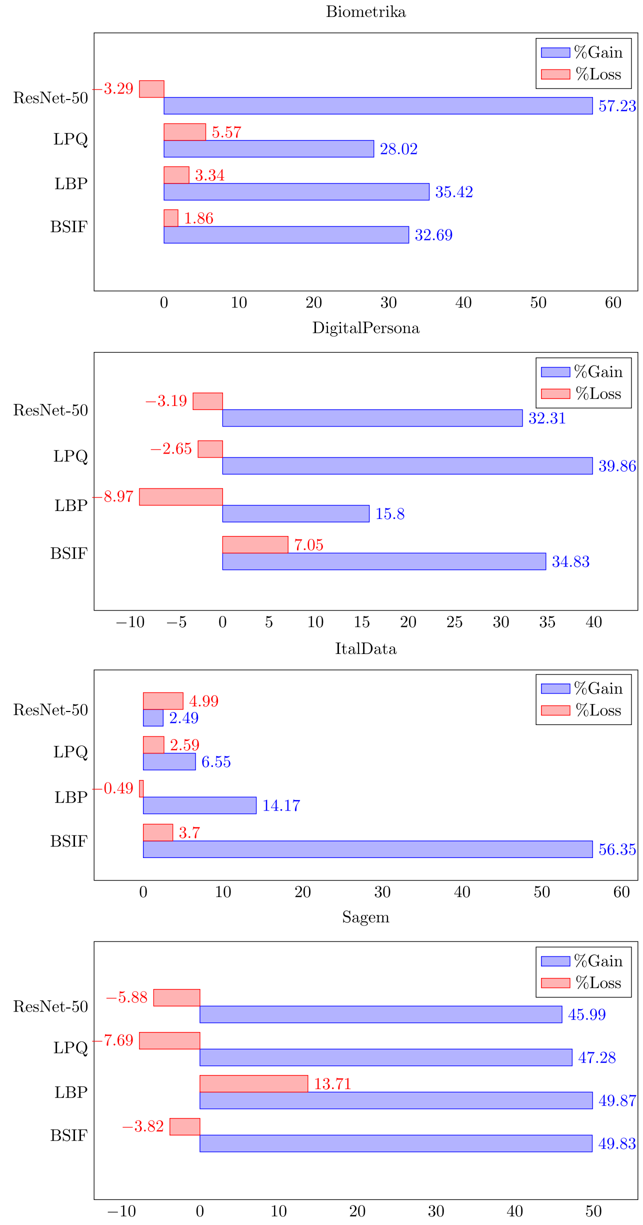


Fig. 3. Comparison of the performance of A-iLearn when used with different features shown on Y axis. Percentage Gain on NF and percentage Loss on KF while learning in second phase are shown on X axis.

On the Digital Persona dataset, the highest performance gain is observed using LPQ features, and the lowest performance loss is observed using LBP features. On average, the performance of LPQ features is the most adequate on this dataset.

We observe the highest performance gain using BSIF features and the lowest performance loss by using LBP features on the Ital Data dataset. Overall, the performance of BSIF features is the most adequate.

We observe the highest gain on the Sagem dataset while using LBP features and the lowest loss while using LPQ features. The performance

Table 3
Stability–Plasticity calculation on LivDet 2011 (Yambay et al., 2012) dataset.

DataSet	A-iLearn											
	I. KF			I. NF			II. KF			II. NF		
	Acc (%)	BPCER (0–1)	APCER (0–1)	Acc (%)	BPCER (0–1)	APCER (0–1)	Acc (%)	BPCER (0–1)	APCER (0–1)	Acc (%)	BPCER (0–1)	APCER (0–1)
Bio	80.23	0.15	0.31	55.39	N/A	0.45	78.87	0.4	0.13	76.18	N/A	0.23
Bio-RT	80.23	0.15	0.31	55.39	N/A	0.45	84.58	0.12	0.22	67.56	N/A	0.32
Dig	89.24	0.02	0.31	26.46	N/A	0.73	90.88	0.07	0.11	35.02	N/A	0.64
Dig-RT	89.24	0.02	0.31	26.46	N/A	0.73	90.72	0.02	0.26	33.51	N/A	0.66
Ital	80.45	0.11	0.37	49.35	N/A	0.51	78.16	0.18	0.31	56.7	N/A	0.43
Ital-RT	80.45	0.11	0.37	49.35	N/A	0.51	81.38	0.11	0.36	52.22	N/A	0.46
Sag	81.3	0.14	0.29	35.62	N/A	0.64	83.08	0.22	0.04	52.96	N/A	0.47
Sag-RT	81.3	0.14	0.29	35.62	N/A	0.64	84.31	0.14	0.19	42.23	N/A	0.56

Table 4
Stability–Plasticity calculation on LivDet 2013 (Ghiani et al., 2013)-LivDet 2015 (Mura et al., 2015) dataset.

DataSet	A-iLearn											
	I. KF			I. NF			II. KF			II. NF		
	Acc (%)	BPCER (0–1)	APCER (0–1)	Acc (%)	BPCER (0–1)	APCER (0–1)	Acc (%)	BPCER (0–1)	APCER (0–1)	Acc (%)	BPCER (0–1)	APCER (0–1)
2013-Bio	98.03	0.02	0.02	93.82	N/A	0.06	97.6	0.03	0	96.52	N/A	0.03
2013-Ital	97.8	0	0.07	84.23	N/A	0.16	95.10	0	0.07	84.25	N/A	0.16
2015-Bio	83.26	0.15	0.03	89.4	N/A	0.11	81.34	0.27	0	93.4	N/A	0.07
2015-Dig	83.64	0.22	0.06	89.37	N/A	0.11	81.33	0.25	0.05	90.4	N/A	0.1

of ResNet is close to LPQ, but on average, LPQ is the most suited for this dataset. Similar observations were obtained for the corresponding features on LivDet 2013 (Ghiani et al., 2013) and LivDet 2015 (Mura et al., 2015); therefore, we did not include them for the sake of space.

5.5. Comparison with state-of-the-art

In this section, we compare the performance of A-iLearn with the current state-of-the-art. To evaluate the performance of A-iLearn along with the existing work in incremental setting, we compare BPCER of A-iLearn on New Fake (NF) and Known Fake (KF) mentioned in the tables given in Section 5.3 with *ferrfake when ferrlive=10%* from Kho et al. (2019) and the results of Rattani et al. (2015) mentioned in the paper. The comparison results on LivDet2011 datasets are given in Table 5. As can be seen from Table 5, performance is evaluated based on two metrics: percentage loss in FPR (or *ferrfake*) on NF and percentage change in FPR on KF. We emphasize that it is essential for an incremental learning algorithm to have a decent percentage loss in FPR on NF, and the percentage change in FPR on KF must always be negative. As A-iLearn makes use of three handcrafted features and one type of deep feature, we report the best performance yielded by any type of feature (indicated in parentheses). It is evident from Table 5, that none of the state-of-the-art models produces all negative values in percentage change in FPR on KF, but A-iLearn does that while maintaining a good percentage loss in FPR on NF. We observed similar behaviour on LivDet 2013 (Ghiani et al., 2013) and LivDet 2015 (Mura et al., 2015), therefore, those values are not reported.

The significance test (independent t-test) conducted with (Kho et al., 2019) and A-iLearn shows that A-iLearn yields significantly better percentage loss in FPR on New Fake, t(6) = 2.54, p = 0.021. Here, y in t(x)= y represents the t-value and x represents the degree of freedom. Similarly, concerning the percentage change in FPR on Known Fake, A-iLearn is significantly better with t(6) = 5.07, p = .001. Similarly, we observed that A-iLearn performs significantly better than (Rattani et al., 2015) feature level fusion, i.e. in terms of percentage loss in FPR on New Fake, t(6) = 2.07, p = 0.04, and in terms of percentage change in FPR on Known Fake, t(6) = 7.41, p = 0.0001. In comparison with (Rattani et al., 2015) score-level fusion, A-iLearn performs significantly better in terms of percentage change in FPR on Known Fake, t(6) = 4.02, p = .003, but not in terms of percentage loss in FPR on New Fake, t(6) = 1.35, p = .11.

5.6. Discussion on results

We describe the results for A-iLearn by the Tables 3–4. We conduct the experiments with the motivation of exploiting the incremental ability of A-iLearn. For that, our emphasis is on reporting the stability and plasticity of the proposed model. We highlight the performance degradation on the known spoof fingerprints and the performance improvement on the new spoof fingerprints before and after learning the new fake data. Our experimental results justify our motivation as we are able to achieve high plasticity with no or negligible loss in stability. As a baseline, we consider a model that requires retraining using the entire data in the second learning phase. Ideally, such a model must not encounter a performance drop on Known Fake in the second phase and then there must be a significant performance improvement on New Fake. As we compare the results of A-iLearn with and without retraining the model with entire data, our results without retraining seem to satisfy the paper's motivation.

In some cases, we get better stability but lesser plasticity while using a particular feature and vice-versa. In some cases, we get reasonable stability and plasticity, but the model's overall accuracy is not adequate with that feature. This kind of result supports the famous no-free-lunch theorem (Wolpert & Macready, 1997), which states that no one model works best for every problem. Therefore, it is advised in machine learning to try multiple models with different settings and find one that works best for a particular problem. A-iLearn performs reasonably well on all the datasets with high-performance gain on New Fake and low/negligible performance loss on Known Fake. In addition, the overall accuracy of the proposed model is also high, with the highest accuracy approaching 96.52% on New Fake.

5.6.1. Performance of A-iLearn in comparison with baseline and state-of-the-art

A-iLearn performs well both in respect of stability and plasticity. Table 3 shows that A-iLearn yields better plasticity than the baseline where we retrain the model from scratch. A-iLearn without retraining the model gives better stability in two of the four cases of LivDet2011 (Yambay et al., 2012) datasets, which encourages the motivation for incremental learning. In addition, the performance on LivDet2013 (Ghiani et al., 2013) and LivDet2015 (Mura et al., 2015) is reasonable well with the highest accuracy reaching 96.52% on New Fake and 97.6% on Known Fake.

Table 5
Performance evaluation of A-iLearn in comparison to the state-of-the-art (Kho et al., 2019; Rattani et al., 2015) on LivDet2011 (Yambay et al., 2012) datasets. In this table, FPR, NF and KF denotes false positive rate, new fake and known false, respectively.

Dataset	Kho et al. (2019)		Feature level Rattani et al. (2015)		Score level Rattani et al. (2015)		A-iLearn	
	%loss in FPR on NF	%change in FPR on KF	%loss in FPR on NF	%change in FPR on KF	%loss in FPR on NF	%change in FPR on KF	%loss in FPR on NF	%change in FPR on KF
Biometrika	62.72	−10.50	64.20	4.44	70.27	−25.48	53.85(ResNet)	−74.36
DigitalPersona	93.95	57.94	89.15	37.78	72.20	91.29	25.86(BSIF)	−57.14
ItalData	67.90	−7.54	55.55	3.18	39.32	19.88	60.00(ResNet)	−100
Sagem	89.49	44.61	89.49	39.66	89.09	13.87	60.00(LBP)	−86.67

5.6.2. Performance of A-iLearn using various features

As described in Section 5.4, A-iLearn performs well on every type of feature. This feature level comparison provides useful insights on the performance of handcrafted features and deep features. Most often, it is argued that the deep features outperform handcrafted features in almost every case; therefore handcrafted features must be discarded. On the contrary, this study proves that handcrafted features give a neck to neck competition and, in some cases, outperform the deep features. Therefore, a single type of feature cannot be trusted to perform well in every case.

6. Conclusions

An incremental learning algorithm must be able to learn from the newly added data while retaining the already acquired knowledge from the past data. We propose a novel incremental learning model A-iLearn and show its working mechanism on spoof fingerprint detection. The proposed algorithm is able to learn the new spoof fingerprints from the current learning phase while maintaining its performance on the "live" and "spoof" fingerprints learned in the previous learning phase. A-iLearn is an adaptive way of learning as it adapts to the similarity inherently present in the data. Also, A-iLearn is an efficient algorithm as it discards the already seen data and keeps only knowledge extracted from it. By doing so, space can be reused for storing the upcoming data. With this motivation, we conducted our experiments and proved the efficacy of A-iLearn. We highlight the stability and plasticity features of A-iLearn on the LivDet2011, LivDet2013 and LivDet2015 dataset and conclude that the proposed framework improves its performance in the new learning phase by 49.57% on an average, without considerable degradation in the existing knowledge. This study provides critical insights into feature level comparison. As a part of future work, integration of the handcrafted and deep features in A-iLearn will be investigated at feature and score level (Rattani et al., 2015) for further performance enhancement.

CRediT authorship contribution statement

Shivang Agarwal: Conceptualization, Methodology, Software, Writing – original draft. **Ajita Rattani:** Validation, Reviewing and editing. **C. Ravindranath Chowdary:** Visualization, Investigation, Supervision.

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.

References

Agarwal, S., & Chowdary, C. R. (2020). A-stacking and A-bagging: Adaptive versions of ensemble learning algorithms for spoof fingerprint detection. *Expert Systems with Applications*, *146*, Article 113160.

Agarwal, S., & Chowdary, C. R. (2021). Combating hate speech using an adaptive ensemble learning model with a case study on covid-19. *Expert Syst. Appl.*, *185*, 115632. http://dx.doi.org/10.1016/j.eswa.2021.115632.

Agarwal, S., Rattani, A., & Chowdary, C. R. (2021). A comparative study on handcrafted features v/s deep features for open-set fingerprint liveness detection. *Pattern Recognition Letters*, *147*, 34–40. http://dx.doi.org/10.1016/j.patrec.2021.03.032.

Ahonen, T., Hadid, A., & Pietikainen, M. (2006). Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(12), 2037–2041.

Al-rimy, B. A. S., Maarof, M. A., & Shaid, S. Z. M. (2019). Crypto-ransomware early detection model using novel incremental bagging with enhanced semi-random subspace selection. *Future Generation Computer Systems*, *101*, 476–491.

Arthur, D., & Vassilvitskii, S. (2007). k-means++: the advantages of carefull seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on discrete algorithms* (pp. 1027–1035).

Bottou, L., & LeCun, Y. (2003). Large scale online learning. In S. Thrun, L. K. Saul, & B. Schölkopf (Eds.), *Advances in neural information processing systems 16 [neural information processing systems, NIPS 2003, December 8-13, 2003, Vancouver and Whistler, British Columbia, Canada]* (pp. 217–224). MIT Press.

Chan, P. K., & Stolfo, S. J. (1998). Toward scalable learning with non-uniform class and cost distributions: A case study in credit card fraud detection. In *KDD'98, Proceedings of the fourth international conference on knowledge discovery and data mining* (pp. 164–168). AAAI Press.

Chapelle, O., Schölkopf, B., & Zien, A. (Eds.), (2006). *Semi-supervised learning*. The MIT Press, http://dx.doi.org/10.7551/mitpress/9780262033589.001.0001.

Chen, Y., & Hao, Y. (2017). A feature weighted support vector machine and K-nearest neighbor algorithm for stock market indices prediction. *Expert Systems with Applications*, *80*, 340–355.

Chowdary, C. R., & Kumar, P. S. (2008). An incremental summary generation system. In G. Das, N. L. Sarda, & P. K. Reddy (Eds.), *Proceedings of the 14th international conference on management of data, December 17-19, 2008, IIT Bombay, Mumbai, India* (pp. 83–92). Computer Society of India / Allied Publishers.

Ditzler, G., & Polikar, R. (2013). Incremental learning of concept drift from streaming imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, *25*(10), 2283–2301. http://dx.doi.org/10.1109/TKDE.2012.136.

Elwell, R., & Polikar, R. (2011). Incremental learning of concept drift in nonstationary environments. *IEEE Transactions on Neural Networks*, *22*(10), 1517–1531.

Gepperth, A., & Hammer, B. (2016). Incremental learning algorithms and applications. In *24th European symposium on artificial neural networks, ESANN 2016, Bruges, Belgium, April 27-29, 2016*.

Ghiani, L., Marcialis, G. L., & Roli, F. (2012). Fingerprint liveness detection by local phase quantization. In *Proceedings of the 21st international conference on pattern recognition, ICPR 2012, Tsukuba, Japan, November 11-15, 2012* (pp. 537–540). IEEE Computer Society.

Ghiani, L., Yambay, D., Mura, V., Tocco, S., Marcialis, G. L., Roli, F., & Schuckers, S. (2013). LivDet 2013 fingerprint liveness detection competition 2013. In J. Fiérrez, A. Kumar, M. Vatsa, R. N. J. Veldhuis, & J. Ortega-Garcia (Eds.), *International conference on biometrics, ICB 2013, 4-7 June, 2013, Madrid, Spain* (pp. 1–6). IEEE.

Gragnaniello, D., Poggi, G., Sansone, C., & Verdoliva, L. (2015). An investigation of local descriptors for biometric spoofing detection. *IEEE Transactions on Information Forensics and Security*, *10*(4), 849–863. http://dx.doi.org/10.1109/TIFS.2015.2404294.

Gu, B., Quan, X., Gu, Y., Sheng, V. S., & Zheng, G. (2018). Chunk incremental learning for cost-sensitive hinge loss support vector machine. *Pattern Recognition*, *83*, 196–208.

Gu, B., Sheng, V. S., & Li, S. (2015). Bi-parameter space partition for cost-sensitive SVM. In Q. Yang, & M. J. Wooldridge (Eds.), *Proceedings of the twenty-fourth international joint conference on artificial intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015* (pp. 3532–3539). AAAI Press, URL: http://ijcai.org/proceedings/2015.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: An update. *SIGKDD Explorations Newsletter*, *11*(1), 10–18. http://dx.doi.org/10.1145/1656274.1656278, URL: http://doi.acm.org/10.1145/1656274.1656278.

He, H., & Chen, S. (2008). IMORL: Incremental multiple-object recognition and localization. *IEEE Transactions on Neural Networks*, *19*(10), 1727–1738. http://dx.doi.org/10.1109/TNN.2008.2001774.

He, J., Mao, R., Shao, Z., & Zhu, F. Incremental learning in online scenario. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE conference on computer vision and pattern recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016* (pp. 770–778). IEEE Computer Society, http://dx.doi.org/10.1109/CVPR.2016.90.

hui Hou, W., kang Wang, X., yu Zhang, H., qiang Wang, J., & Li, L. (2020). A novel dynamic ensemble selection classifier for an imbalanced data set: An application for credit risk assessment. *Knowledge-Based Systems*, *208*, Article 106462. http://dx.doi.org/10.1016/j.knosys.2020.106462.

Iwashita, A. S., de Albuquerque, V. H. C., & Papa, J. P. (2019). Learning concept drift with ensembles of optimum-path forest-based classifiers. *Future Generation Computer Systems*, *95*, 198–211.

Jia, S., Guo, G., Xu, Z., & Wang, Q. (2020). Face presentation attack detection in mobile scenarios: A comprehensive evaluation. *Image and Vision Computing*, *93*, Article 103826. http://dx.doi.org/10.1016/j.imavis.2019.11.004.

Jia, X., Yang, X., Cao, K., Zang, Y., Zhang, N., Dai, R., Zhu, X., & Tian, J. (2014). Multi-scale local binary pattern with filters for spoof fingerprint detection. *Information Sciences*, *268*, 91–102, New Sensing and Processing Technologies for Hand-based Biometrics Authentication.

Jin, C., Li, S., Kim, H., & Park, E. (2011). Fingerprint liveness detection based on multiple image quality features. In Y. Chung, & M. Yung (Eds.), *Information security applications* (pp. 281–291). Berlin, Heidelberg: Springer Berlin Heidelberg.

Kannala, J., & Rahtu, E. (2012). BSIF: Binarized statistical image features. In *Proceedings of the 21st international conference on pattern recognition, ICPR 2012, Tsukuba, Japan, November 11-15, 2012* (pp. 1363–1366). IEEE Computer Society.

Kho, J. B., Lee, W., Choi, H., & Kim, J. (2019). An incremental learning method for spoof fingerprint detection. *Expert Systems with Applications*, *116*, 52–64.

Li, Z., Huang, W., Xiong, Y., Ren, S., & Zhu, T. (2020). Incremental learning imbalanced data streams with concept drift: The dynamic updated ensemble algorithm. *Knowledge-Based Systems*, *195*, Article 105694.

Lu, Y., Boukharouba, K., Boonært, J., Fleury, A., & Lecœuche, S. (2014). Application of an incremental SVM algorithm for on-line human recognition from video surveillance using texture and color features. *Neurocomputing*, *126*, 132–140, Recent trends in Intelligent Data Analysis Online Data Processing.

Luo, X., Xia, Y., & Zhu, Q. (2012). Incremental collaborative filtering recommender based on regularized matrix factorization. *Knowledge-Based Systems*, *27*, 271–280. http://dx.doi.org/10.1016/j.knosys.2011.09.006.

Marasco, E., & Ross, A. (2014). A survey on antispoofing schemes for fingerprint recognition systems. *ACM Computing Surveys*, *47*(2), 28:1–28:36. http://dx.doi.org/10.1145/2617756, URL: http://doi.acm.org/10.1145/2617756.

Melville, P., & Mooney, R. J. (2005). Creating diversity in ensembles using artificial data. *Information Fusion*, *6*(1), 99–111. http://dx.doi.org/10.1016/j.inffus.2004.04.001, Diversity in Multiple Classifier Systems.

Menotti, D., Chiachia, G., Pinto, A., Schwartz, W. R., Pedrini, H., Falcão, A. X., & Rocha, A. (2015). Deep representations for iris, face, and fingerprint spoofing detection. *IEEE Transactions on Information Forensics and Security*, *10*(4), 864–879. http://dx.doi.org/10.1109/TIFS.2015.2398817.

Muhlbaier, M. D., Topalis, A., & Polikar, R. (2009). Learn^{++} .NC: Combining ensemble of classifiers with dynamically weighted consult-and-vote for efficient incremental learning of new classes. *IEEE Transactions on Neural Networks*, *20*(1), 152–168.

Mura, V., Ghiani, L., Marcialis, G. L., Roli, F., Yambay, D. A., & Schuckers, S. A. C. (2015). LivDet 2015 fingerprint liveness detection competition 2015. In *IEEE 7th international conference on biometrics theory, applications and systems, BTAS 2015, Arlington, VA, USA, September 8-11, 2015* (pp. 1–6). IEEE.

Nallaperuma, D., Nawaratne, R., Bandaragoda, T., Adikari, A., Nguyen, S., Kempitiya, T., De Silva, D., Alahakoon, D., & Pothuhera, D. (2019). Online incremental machine learning platform for big data-driven smart traffic management. *IEEE Transactions on Intelligent Transportation Systems*, *20*(12), 4679–4690. http://dx.doi.org/10.1109/TITS.2019.2924883.

Nogueira, R. F., de Alencar Lotufo, R., & Campos Machado, R. (2016). Fingerprint liveness detection using convolutional neural networks. *IEEE Transactions on Information Forensics and Security*, *11*(6), 1206–1213.

Ojansivu, V., & Heikkilä, J. (2008). Blur insensitive texture classification using local phase quantization. In A. Elmoataz, O. Lezoray, F. Nouboud, & D. Mammass (Eds.), *Image and signal processing* (pp. 236–243). Berlin, Heidelberg: Springer Berlin Heidelberg.

Pan, S. J., Tsang, I. W., Kwok, J. T., & Yang, Q. (2011). Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, *22*(2), 199–210. http://dx.doi.org/10.1109/TNN.2010.2091281.

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1345–1359. http://dx.doi.org/10.1109/TKDE.2009.191.

Peng, H., Bao, M., Li, J., Bhuiyan, M. Z. A., Liu, Y., He, Y., & Yang, E. (2018). Incremental term representation learning for social network analysis. *Future Generation Computer Systems*, *86*, 1503–1512.

Platt, J. (1998). Fast training of support vector machines using sequential minimal optimization. In *Advances in kernel methods - support vector learning* (pp. 185–208). MIT Press.

Polikar, R., Upda, L., Upda, S. S., & Honavar, V. (2001). Learn++: an incremental learning algorithm for supervised neural networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *31*(4), 497–508. http://dx.doi.org/10.1109/5326.983933.

Pozzolo, A. D., Caelen, O., Borgne, Y. A. L., Waterschoot, S., & Bontempi, G. (2014). Learned lessons in credit card fraud detection from a practitioner perspective. *Expert Systems with Applications*, *41*(10), 4915–4928.

Rattani, A., Scheirer, W. J., & Ross, A. (2015). Open set fingerprint spoof detection across novel fabrication materials. *IEEE Transactions on Information Forensics and Security*, *10*(11), 2447–2460. http://dx.doi.org/10.1109/TIFS.2015.2464772.

Rigollet, P. (2007). Generalization error bounds in semi-supervised classification under the cluster assumption. *Journal of Machine Learning Research*, *8*, 1369–1392, URL: http://dl.acm.org/citation.cfm?id=1314545.

Roy, D., Panda, P., & Roy, K. (2020). Tree-CNN: A hierarchical deep convolutional neural network for incremental learning. *Neural Networks*, *121*, 148–160.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. S., Berg, A. C., & Li, F. (2015). ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, *115*(3), 211–252. http://dx.doi.org/10.1007/s11263-015-0816-y.

Saha, A., & Sindhwani, V. (2012). Learning evolving and emerging topics in social media: A dynamic nmf approach with temporal regularization. In *WSDM '12, Proceedings of the fifth ACM international conference on web search and data mining* (pp. 693–702). New York, NY, USA: ACM.

Saveriano, M., An, S., & Lee, D. (2015). Incremental kinesthetic teaching of end-effector and null-space motion primitives. In *IEEE international conference on robotics and automation, ICRA 2015, Seattle, WA, USA, 26-30 May, 2015* (pp. 3570–3575). IEEE.

Shan, G., Xu, S., Yang, L., Jia, S., & Xiang, Y. (2020). Learn#: A novel incremental learning method for text classification. *Expert Systems with Applications*, *147*, Article 113198.

Sharma, R. P., & Dey, S. (2019). Two-stage quality adaptive fingerprint image enhancement using fuzzy C-means clustering based fingerprint quality analysis. *Image and Vision Computing*, *83–84*, 1–16. http://dx.doi.org/10.1016/j.imavis.2019.02.006.

Stojanov, S., Mishra, S., Thai, N. A., Dhanda, N., Humayun, A., Yu, C., Smith, L. B., & Rehg, J. M. (2019). Incremental object learning from contiguous views. In *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 8769–8778).

Tharwat, A., & Schenck, W. (2020). Balancing exploration and exploitation: A novel active learner for imbalanced data. *Knowledge-Based Systems*, *210*, Article 106500. http://dx.doi.org/10.1016/j.knosys.2020.106500.

Wei, X., Liu, S., Xiang, Y., Duan, Z., Zhao, C., & Lu, Y. (2020). Incremental learning based multi-domain adaptation for object detection. *Knowledge-Based Systems*, *210*, Article 106420. http://dx.doi.org/10.1016/j.knosys.2020.106420.

Win, K. N., Li, K., Chen, J., Viger, P. F., & Li, K. (2020). Fingerprint classification and identification algorithms for criminal investigation: A survey. *Future Generation Computer Systems*, *110*, 758–771.

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, *1*(1), 67–82. http://dx.doi.org/10.1109/4235.585893.

Wu, Y., Chen, Y., Wang, L., Ye, Y., Liu, Z., Guo, Y., & Fu, Y. (2019). Large scale incremental learning. In *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 374–382).

Yambay, D., Ghiani, L., Denti, P., Marcialis, G. L., Roli, F., & Schuckers, S. A. C. (2012). Livdet 2011 - fingerprint liveness detection competition 2011. In A. K. Jain, A. Ross, S. Prabhakar, & J. Kim (Eds.), *5th IAPR international conference on biometrics, ICB 2012, New Delhi, India, March 29 - April 1, 2012* (pp. 208–215). IEEE.

Yang, Z., Al-Dahidi, S., Baraldi, P., Zio, E., & Montelatici, L. (2020). A novel concept drift detection method for incremental learning in nonstationary environments. *IEEE Transactions on Neural Networks and Learning Systems*, *31*(1), 309–320.

Yu, C., Yao, C., Pei, M., & Jia, Y. (2019). Diffusion-based kernel matrix model for face liveness detection. *Image and Vision Computing*, *89*, 88–94. http://dx.doi.org/10.1016/j.imavis.2019.06.009.

Yuan, X., Wang, R., Zhuang, Y., Zhu, K., & Hao, J. (2018). A concept drift based ensemble incremental learning approach for intrusion detection. In *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)* (pp. 350–357).

Zhang, J., Zhang, J., Ghosh, S., Li, D., Tasci, S., Heck, L. P., Zhang, H., & Kuo, C. J. (2020). Class-incremental learning via deep model consolidation. In *IEEE winter conference on applications of computer vision, WACV 2020, Snowmass Village, CO, USA, March 1-5, 2020* (pp. 1120–1129). IEEE, http://dx.doi.org/10.1109/WACV45572.2020.9093365.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., & He, Q. (2021). A comprehensive survey on transfer learning. *Proceedings of the IEEE*, *109*(1), 43–76. http://dx.doi.org/10.1109/JPROC.2020.3004555.