Computer Communications 229 (2025) 108006

Contents lists available at ScienceDirect

Computer Communications

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

A dual-tier adaptive one-class classification IDS for emerging cyberthreats

Md. Ashraf Uddin [a,*], Sunil Aryal [a], Mohamed Reda Bouadjenek [a], Muna Al-Hawawreh [a], Md. Alamin Talukder [b]

[a] School of Information Technology, Deakin University, Geelong, VIC 3125, Australia
[b] Department of Computer Science and Engineering, International University of Business Agriculture and Technology, Dhaka, Bangladesh

ARTICLE INFO

Dataset link: https://www.unb.ca/cic/datasets/index.html, https://research.unsw.edu.au/projects/unsw-nb15-dataset

Keywords:
IoT
Network traffic
IDS
Machine learning
Deep learning
Intrusion detection

ABSTRACT

In today's digital age, our dependence on IoT (Internet of Things) and IIoT (Industrial IoT) systems has grown immensely, which facilitates sensitive activities such as banking transactions and personal, enterprise data, and legal document exchanges. Cyberattackers consistently exploit weak security measures and tools. The Network Intrusion Detection System (IDS) acts as a primary tool against such cyber threats. However, machine learning-based IDSs, when trained on specific attack patterns, often misclassify new emerging cyberattacks. Further, the limited availability of attack instances for training a supervised learner and the ever-evolving nature of cyber threats further complicate the matter. This emphasizes the need for an adaptable IDS framework capable of recognizing and learning from unfamiliar/unseen attacks over time. In this research, we propose a one-class classification-driven IDS system structured on two tiers. The first tier distinguishes between normal activities and attacks/threats, while the second tier determines if the detected attack is known or unknown. Within this second tier, we also embed a multi-classification mechanism coupled with a clustering algorithm. This model not only identifies unseen attacks but also uses them for retraining them by clustering unseen attacks. This enables our model to be future-proofed, capable of evolving with emerging threat patterns. Leveraging one-class classifiers (OCC) at the first level, our approach bypasses the need for attack samples, addressing data imbalance and zero-day attack concerns and OCC at the second level can effectively separate unknown attacks from the known attacks. Our methodology and evaluations indicate that the presented framework exhibits promising potential for real-world deployments.

1. Introduction

The growing reliance on IoT/IIoT (Industrial IoT) networks has inadvertently increased the likelihood of new cyberattack occurrences. Conventional IDS struggle to effectively detect modern threats due to their limited ability to adapt and counter the sophisticated techniques used in these attacks [1,2]. Development of a practical IDS for critical applications is still challenging despite the considerable amount of research conducted in this field. Most existing IDS systems do not have the capability of detecting the continuously emerging new security threats and retraining the model using the novel attack samples [3]. The dynamic nature of cyberattacks necessitates regular updates to IDS to effectively detect and respond to emerging attack patterns. The detection of new malicious attacks has garnered significant interest among researchers. Contemporary IDSs are not effective in detecting unseen attacks as those IDSs generate a significant number of false negatives [4,5].

Attackers frequently evade a company's security measures by exploiting vulnerabilities that have not been disclosed or seen by security personnel. These kinds of attacks are termed as zero-day vulnerabilities, a vendor's undetected or unaddressed threat for software or applications. Bilge et al. [6] presented the substantial impacts of zero-day attacks on both consumers and systems. Their research revealed that organizations encounter considerable challenges in detecting zero-day attacks, with an average detection time ranging from 312 days to as long as 30 months. This indicates the critical need for the development of IDS capable of promptly identifying and mitigating such attacks as soon as they attack the system. Most traditional security methods utilize either signatures or machine learning models trained for predefined normal and attack sample patterns to detect vulnerabilities. However, these methods are ineffective in identifying zero-day vulnerabilities due to the uncertainty of signature data. Detecting zero-day exploits using conventional IDS is highly challenging [7] because such vulnerability could potentially remain undisclosed to the general public for an extended period, ranging from several months to several years, owing to the advanced capabilities of the attackers [8].

* Corresponding author.
E-mail addresses: ashraf.uddin@deakin.edu.au (Md.A. Uddin), sunil.aryal@deakin.edu.au (S. Aryal), alamin.cse@iubat.edu (Md.A. Talukder).

https://doi.org/10.1016/j.comcom.2024.108006
Received 1 May 2024; Received in revised form 2 September 2024; Accepted 7 November 2024
Available online 14 November 2024

To improve the performance of IDS by minimizing false-negative rates, it is important for an IDS to effectively and accurately detect new/zero-day attacks. This can be achieved by regularly detecting unknown attacks and updating IDS in real-time [9]. Regarding this, Masdari and Khezri [10] highlighted the importance of adaptability in an IDS for the successful detection of novel threats. An Adaptive IDS refers to a classification model that is dynamically updated to identify emerging attack instances.

The precondition of building an adaptive model is to correctly detect unseen/new attacks and label them for retraining the model. Traditional IDS typically identify new attacks offline through manual or semi-automated processes [11]. This task necessitates significant exertion from experts. Automated models are faster and more efficient compared to manually coded models. An adaptive model facilitates collecting and promptly integrating into current detection models to swiftly mitigate potential harm caused by such previously unseen threats. In recent years, there has been an increased focus on the utilization of data mining and machine learning methods for building intrusion detection models and others [12–14]. The models are developed based on normal behavior and known threats to detect previously unseen threats. In this process, we first require to distinguish attack samples from the normal samples. Next, unseen/unknown attacks are collected and labeled for training a multi-classifier model so that it can identify such attacks next time. Many modern IDS systems require the availability of extensive, balanced datasets containing both normal and attack instances as they utilize mostly supervised machine learning models. In real-world scenarios, procuring a substantial number of attack instances is difficult, given that malicious network traffic does not occur at the same frequency as regular traffic. In addition, attack patterns evolve swiftly, meaning a supervised model, trained on a specific attack distribution, might fail in recognizing novel or zero-day attacks [15,16]. To address this issue, Some existing studies [17–20] have proposed one-class classification (OCC) methods that train exclusively on normal instances, identifying attack instances as anomalies. Our findings show that this strategy efficiently differentiates between normal and attack instances with higher accuracy. However, when it comes to pinpointing specific attack families—like DoS, Ransomware, Trojan Horse, and Spywar, existing solutions seek multi-class supervised learning. Such models can only identify attack types they have been trained on, leading them to misclassify new attacks as known variants. One potential solution involves examining the prediction probability generated by a multi-class classifier when it assesses an instance's likelihood of belonging to a known attack type. By setting a threshold probability, we might be able to distinguish between known and unknown attacks. Yet, determining an optimal threshold remains a challenge since multi-classifiers often assign a high probability to one of the known attack categories even for unseen attack instances.

In the domain of IDS, a semi-supervised learning technique called a one-class classifier (OCC) is intended to find patterns in data that deviate from the normal behavior of traffic instances [21]. An OCC is trained just on data that represents regular behavior, as opposed to typical classifiers that use both normal and malicious instances to categorize new data points. Because of this, it is especially well-suited for identifying hidden attacks, or zero-day or unique threats that do not correspond with any known attack signatures.

OCC like OCSVM (One Class SVM) has been applied in some recent works [21–24] to detect attack and normal instances. Nevertheless, a limited number of studies [21,24] investigated the SVM or OCSVM model to distinguish unseen attacks from known attack types. While it is more usual in the literature to utilize the OCC model to distinguish between unseen attacks and regular network traffic, there has been very little research done in the literature to distinguish unseen attacks from seen attack categories using either the supervised learner or semi-supervised learning. For security administration purposes, it is necessary to examine the effectiveness of the most recently developed OCC models, such as usfAD, in distinguishing unseen attacks from seen attack categories. It is also necessary to accurately identify the family types of cyberattacks. This can enable administrators for the appropriate respond appropriately to emerging attack types.

Most of the machine learning-based literature has primarily focused on distinguishing normal network traffic from attacks but has not prioritized the categorization of specific attack families or the detection of unseen attacks [24,25]. Consequently, when employing these models, it becomes challenging to generate comprehensive reports on the nature of the attacks. Such reports could provide valuable insights for security administrators to formulate and enforce effective security policies. Another limitation of this approach is its inability to differentiate the behavior of novel attacks from benign or previously known attack categories.

Research in IDS has partially addressed the challenge of unknown network threats. Some studies [17–20] have recognized these threats but have not provided methods for systems to learn and adapt to them. Others [3,26] have outlined retraining processes with new, unseen threats but fell short in the initial detection of such attacks. Literature suggests network administrators manually detect and categorize these novel attacks to enhance supervised learning models. Therefore, there needs a robust, adaptive IDS that not only detects new kinds of attacks at multiple levels but also can incorporate these into its retraining process so that it can anticipate and recognize future threats more effectively.

Soltani et al. [27] suggested deep learning models to identify novel attacks and a semi-supervised clustering method for updating the model. However, the model did not produce promising outcomes and IDS datasets are high-dimensional which demands a significant computational cost for the deployment and retraining of deep learning models. Therefore, the utilization of machine learning (ML) models is the preferred choice for constructing a resilient IDS capable of detecting and retraining unknown attacks.

To counteract this limitation, we introduce a two-staged integrated OCC-based technique for detecting unknown attacks and retraining the model. In this work, we have suggested a two-level hierarchical structure for detecting known and unknown attacks. To the best of our knowledge, our research is the first utilization of a hierarchical structure to construct an IDS that effectively distinguishes between known, unknown, and benign network traffic instances. This novel approach leverages recently developed OCC techniques(usfAD) in the context of network security. Generally, machine learning (ML) models exhibit proficiency in creating distinct decision boundaries for binary classifications, like differentiating between normal and attack traffic. Based on the presumption that initially separating attacks results in higher accuracy compared to concurrently classifying normal, known, and unknown attacks, we propose a hierarchical approach. In this approach, the first level handles the classification of normal and attacks, while the second level specializes in distinguishing between unknown and known attacks. This design enables the model to concentrate specifically on establishing decision boundaries between these two distinct categories of network traffic.

We place a usfAD algorithm at the first level in this structure to distinguish benign and attack samples. At the second level, another usfAD approach is placed to recognize known and unknown attacks. The usfAD at the first level is trained using only normal data whereas the usfAD at the second level is trained using known attack samples. By training solely on known attack types, our method can effectively categorize unfamiliar attacks as anomalies. We also advocate for a parallel supervised learning model, trained on known attack types. When the OCC method tags an instance as a known attack, the supervised model then identifies its specific family. If the OCC labels an attack instance as unknown, the instance is set aside in a designated bucket. Once we accumulate a significant number of such unknown instances, a clustering technique (DBSCAN/DPC) helps group similar unknown attack types. We then consider the largest cluster which might consist of several unknown attack samples depending on the purity of

the cluster. We extract the samples with dominating unknown attack types to retrain, both the supervised model and the second-level OCC to ensure future attacks of this nature are correctly detected. Finally, We rigorously trained and assessed our proposed IDS system using ten distinct IDS datasets. This evaluation is executed using stratified 5-fold cross-validation to ensure its robustness. We measured the model's effectiveness by measuring average accuracy, precision, recall, and the f1-score.

The structure of this paper is as follows: Section 2 presents a review of related literature. Section 3 details the hierarchical architecture and methodology of the adaptive model. In Section 4, we present the results of our experiments and analyze the outcome to show the effectiveness of the proposed model. Finally, Section 5 summarizes the paper and outlines potential future research directions.

2. Literature review

In recent years, IDSs have been proposed, focusing on both traditional and machine learning-based methods. Many studies have explored the use of deep learning architectures, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), to enhance detection accuracy. Additionally, hybrid approaches that combine anomaly detection and signature-based techniques have gained attention for their ability to address both known and unknown threats. Despite these advancements, challenges remain in improving detection rates while minimizing false positives and ensuring scalability in real-world environments.

Critical evaluation of existing literature

Roshan et al. [3] aimed to improve IDS adaptability using Extreme Learning Machines (ELM), effectively detecting both known and unknown threats. While their approach included an efficient update mechanism for new data patterns, the reliance on human experts for updates could limit scalability in high-traffic environments. This underscores a gap in automating the updating process in IDS, a key area our work seeks to address.

Al-Yaseena et al. [21] presented a real-time detection system using multi-agent strategies combined with hybrid SVM and ELM models. Although this system showed efficiency in reducing training costs and demonstrated superior performance on the KDDCup'99 dataset, the study's reliance on traditional datasets limits its applicability to real-world scenarios, particularly in cloud environments. This indicates a gap in evaluating IDS on more current and diverse datasets, a gap our research intends to fill.

Singh et al. [22] proposed a framework for detecting zero-day attacks using a probabilistic methodology. While their approach effectively identified new vulnerabilities, the framework's hybrid nature introduces complexity, which may hinder real-time deployment. Addressing the balance between complexity and real-time applicability remains a critical challenge in IDS development.

Sethi et al. [28] developed an adaptive IDS based on a Double Deep Q-Network (DDQN) for cloud environments. Their use of both ISOT-CID and NSL-KDD datasets provided a more comprehensive evaluation, yet the system's adaptability to emerging threats still relied on predefined categories. This highlights the need for more dynamic approaches that can continuously learn and adapt without relying solely on predefined attack patterns.

Soltani et al. [27] introduced a deep learning-based framework for adaptive IDS, focusing on zero-day attacks using a combination of DOC++ and clustering techniques. Although effective, their model still required manual intervention for labeling, a process that could be automated for greater efficiency. Our approach seeks to further reduce human dependency by implementing a fully automated hierarchical structure for detecting and categorizing attacks.

Hindy et al. [24] utilized autoencoders for zero-day attack detection, achieving high recall rates. However, the model's high reliance on autoencoders may not generalize well across different types of attacks, especially as new, more complex threats emerge. This raises a need for more robust methods that can handle a broader range of anomalies, a gap our research aims to address.

Al-Zewairi et al. [23] explored unknown attack classification using shallow and deep neural networks, highlighting the generalization errors present in current IDS models. Their findings underscore the necessity for more innovative approaches to reduce these errors, a challenge we aim to tackle through our advanced semi-supervised approach.

Xianwei et al. [29] proposed an ensemble-based adaptive IDS using decision trees and other base classifiers, showing promise in improving detection performance. However, their approach's effectiveness in real-time scenarios remains unclear, particularly concerning the computational overhead of ensemble methods. This highlights the need for more efficient algorithms that balance performance with real-time applicability.

Mike et al. [30] and Ali et al. [31] provided insights into feature selection and comparative studies of IDS methodologies. However, their reliance on supervised learning limits their ability to detect unseen attacks. Our research addresses this by incorporating unsupervised and semi-supervised learning techniques to improve detection accuracy for unknown threats.

Ahmet et al. [32] and Elfeshawy et al. [33] explored alternative approaches, such as social media analysis and adaptive neural networks, respectively. While innovative, these methods still face challenges in scalability and adaptability, which are crucial for modern IDS.

From this critical evaluation, it is evident that while significant advancements have been made in IDS research, gaps remain in automating the adaptation process, reducing human intervention, and improving detection accuracy for zero-day attacks. Our work addresses these challenges by proposing an advanced semi-supervised IDS model that leverages a hierarchical structure, offering a more efficient and scalable solution for real-time threat detection.

2.1. Identifying the research gap

A thorough evaluation of existing literature reveals several critical gaps in current Intrusion Detection Systems (IDS). Firstly, many IDS models struggle with adaptability to new attack patterns, facing high computational costs and difficulties with frequent retraining using labeled data. This results in limited effectiveness in real-time applications. Additionally, these systems often rely heavily on human expertise for updating attack signatures, which is impractical in dynamic environments where rapid detection is essential. Another significant issue is the inadequacy of current techniques like One-Class SVM and Local Outlier Factor in effectively detecting zero-day attacks, often leading to high false-negative rates. Furthermore, while some models utilize binary or multi-classification methods, the potential of hierarchical structures to enhance detection accuracy remains underexplored. Many studies also rely on traditional datasets like NSL-KDD, which may not accurately represent real-world network complexities, limiting the practical deployment of these models. Lastly, the use of semi-supervised learning in IDS is limited, despite its potential to leverage both labeled and unlabeled data for improved threat detection.

2.2. Proposing the hypothesis

To address these identified gaps, we propose a novel IDS framework that integrates a hierarchical structure, semi-supervised learning, and clustering techniques. Our approach introduces a two-level hierarchical model where the first level distinguishes benign traffic from attacks, and the second level differentiates between known and unknown attacks. This sequential filtering enhances detection accuracy by focusing

on specific threat categories. We employ the semi-supervised model usfAD, which uses both labeled and unlabeled data to adapt to new threats without requiring prior knowledge of unknown attack samples. Additionally, we integrate clustering techniques such as DBSCAN to group similar unknown attack samples, facilitating more effective retraining of the IDS. This comprehensive approach will be rigorously tested using diverse, real-world datasets to ensure robustness and practical applicability, aiming to provide a more adaptive and scalable solution for IDS challenges.

2.3. Posing the research question

Based on the proposed hypothesis of integrating a two-tier architecture with semi-supervised learning and clustering techniques, the relevant research question can be formulated as follows: **"How can a two-tier IDS framework, incorporating the usfAD semi-supervised model and clustering methods, improve the detection of both known and unknown network attacks while minimizing false negatives and computational costs?"** This question addresses the core aim of evaluating the effectiveness of the proposed IDS framework in enhancing detection accuracy and adaptability, particularly in distinguishing between known and unknown attacks, and its efficiency in practical, real-world scenarios.

2.4. Explaining the proposed method

To answer this research question, we propose a multi-faceted approach:

1. Two-Tier Architecture: We implement a two-tier IDS framework where the first tier uses usfAD to classify traffic into benign and attack categories. The second tier employs another usfAD model to further differentiate between known and unknown attacks. This architecture streamlines the classification process, improving overall accuracy and efficiency by focusing on specific types of traffic.
2. Semi-Supervised Learning: We utilize the usfAD semi-supervised model at both tiers of the architecture. The model is trained on labeled data for known attacks and unlabeled data for unknown attacks. This method allows the IDS to adapt to new threats without requiring extensive labeled datasets, addressing the challenge of detecting zero-day attacks.
3. Clustering Techniques: To handle unknown attacks effectively, we incorporate clustering algorithms such as DBSCAN. These algorithms group similar unknown attack samples, facilitating the identification of attack patterns and improving the model's ability to recognize and classify new attack types. The largest clusters are used to retrain the IDS, ensuring that the system continually updates its capabilities.
4. Evaluation and Validation: We rigorously test the proposed IDS framework using a variety of real-world datasets to assess its performance. Metrics such as accuracy, precision, recall, and F1-score are used to evaluate the system's effectiveness in detecting both known and unknown attacks, while also measuring computational efficiency.

This comprehensive method aims to provide a robust, adaptable, and efficient IDS solution, addressing the limitations identified in existing systems and enhancing overall threat detection capabilities.

3. Methodology of adaptive hierarchical model

In this work, we aim to ensure that future instances of novel attacks are correctly recognized and classified by the IDS, fostering a more resilient and adaptive cybersecurity infrastructure. We introduce a dual-layered framework for the IDS. At the first level, we employ an OCC detection method to discern between benign and malicious network instances. In the next tier, another OCC mechanism categorizes detected attack samples as either known or unknown. Within this level, a supervised algorithm classifies the family of known attacks, while unknown attacks are stored separately. These unknown attacks are then grouped using clustering, and their respective groups are labeled with the assistance of expert judgment. Figs. 1 visually represent the comprehensive structure of our IDS framework, which we detail in the sections below.

1.1 We collect ten different publicly accessible benchmarks IDS datasets: NSL-KDD, UNSW-NB15, CIC-DoS2017, CIC-DDoS2019, Darknet 2020, MalMem 2022, XIIOTID, ToN-IoT-Network, ToN-IoT-Linux and ISCXURL2016.
1.2 The pre-processing unit performs minimal pre-processing including converting categorical values to numerical values using label encoding and standardizing the features' value using min–max techniques.
1.3 We apply 5-fold stratified cross-validation for training and testing the model for every dataset. In every fold of the stratified cross-validation, the dataset is split into training and testing sets.

1.4 To train the OCC1 (usfAD, Local Outlier Factor (LOF), Isolation Forest (IOF), and Auto Encoder(AE)) at the first level, we extract the normal instances from the training set.
2.1 [2.2][2.3] To train the OCC2 (usfAD, LOF, IOF, and AE) at the second level, we extract only attack instances from the training datasets. In this case, we deliberately create different combinations of unknown and known attacks. If the attack categories in the training set are $a_1, a_2, a_3, and a_4$. We create the following combination of unknown and known categories: $[a_1], [a_2, a_3, a_4]$ ([unknown], [known], $[a_1]$ is unknown set and $[a_2, a_3, a_4]$ is known attack categories), $([a_2], [a_1, a_3, a_4])$, $([a_3], [a_1, a_2, a_4])$, $([a_4], [a_1, a_2, a_3])$, $([a_1, a_2], [a_3, a_4])$, $([a_1, a_3], [a_2, a_4])$, $([a_2, a_3], [a_1, a_4])$, $([a_1, a_2, a_3], [a_4])$ and so on. We train the second OCC2 and the supervised learner (M_1) using the known attacks from each set of unknown and known attack combinations.
2.4 In the testing phase, for the usfAD model, we set a threshold to differentiate between normal and attack instances. We obtain scores after training the usfAD for training and testing instances. The formula for the threshold is TH = mean(scores of training instances)- 3* standard deviation (scores of training instances). If the score of a testing instance is less than the threshold, we consider the instance as an attack, otherwise, it is labeled as a normal instance. For all the OCC models including usfAD, we collect all the instances that are predicted as attacks by the OCC1 as a testing set for the second OCC2 model.
2.5 If a data point is predicted as an attack class from the OCC1 model, then this is predicted by the OCC2 to identify if the attack point is known or unknown. If an instance is identified as a known attack, this is passed to the supervised model (M_1) at the second level to reveal the specific attack family type.
2.6 After collecting a certain number of unknown attacks, we utilize DBSCAN and DPC clustering to group similar kinds of unknown attack samples. The number of similar groups in the unknown samples is not known. For this reason, we choose DBSCAN/DPC as we do not need to determine the number of clusters in DBSCAN/DPC.
2.7 In our methodology, for both known and novel attack scenarios, we focus on the largest cluster identified in the data, which likely includes samples from various attack classes. Specifically, we target the predominant unknown attack type within this cluster for retraining our second-level OCC and Random Forest models (RF). This enables the models to better identify such attacks in future instances. The training set does not contain particular types of attacks. In our simulation, we ensure that the test set includes all kinds of attack instances. In an ideal case, our second-level

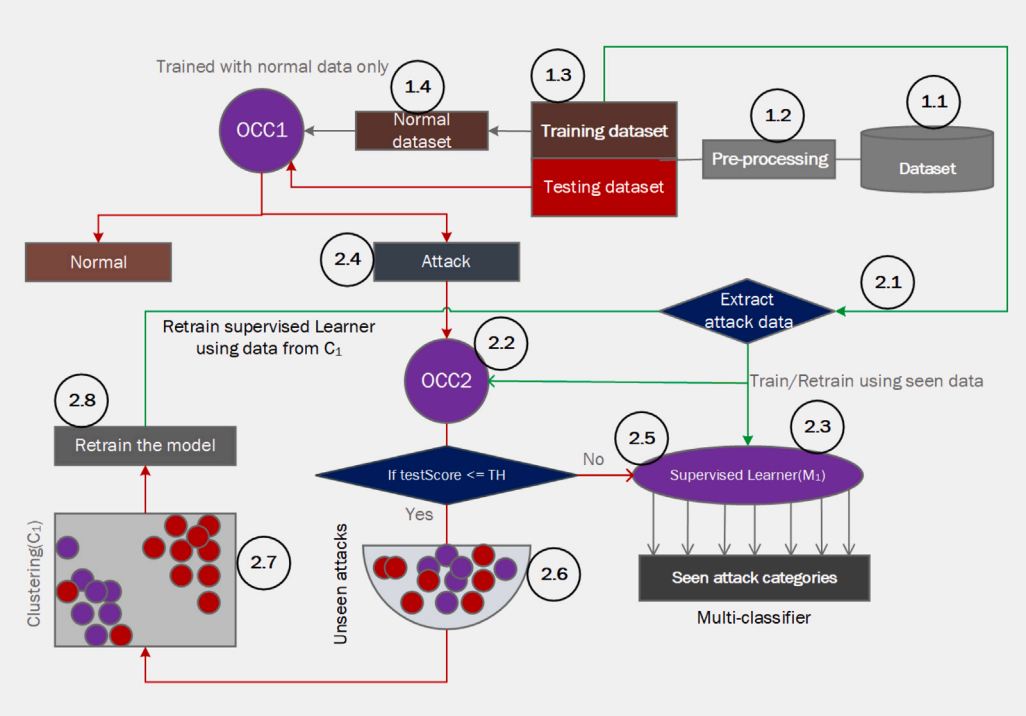


Fig. 1. Overview of adaptive IDS framework.

OCC detects such attack samples that are not used in the training set as unknown attacks.

Upon accumulating a sufficient quantity of such unknown samples (e.g., 1000), we apply clustering algorithms like DBSCAN to these samples. For instance, if the majority cluster corresponds to attack 1 initially, we retrain OCC2 and RF with this new data. Consequently, in subsequent iterations, the model starts recognizing attack 1. We repeat this process, collecting another set of 1000 unknown samples for retraining with the next prevalent attack type, and so on. This iterative retraining process aims to progressively enhance the model's capability to autonomously learn and recognize emerging attack categories over time.

2.8 The known attack instances are used to train a supervised model (RF) for identifying individual categories of known attacks. Finally, performance is measured using various performance metrics such as accuracy, precision, recall, and f1-score.

In the next section, we describe the retraining of the proposed model in detail.

3.1. Simulation of retraining semi-supervised and supervised learner

Fig. 3 presents a methodology for updating training processes in the face of unidentified attack categories. Initially, an OCC algorithm at the second level detects an instance as either a known or unknown attack category. Known attacks are forwarded to a RF classifier (M_1) to detect their specific family types such as DoS, Ransomware, or Trojan Horse. On the contrary, unknown attacks are temporarily stored in a bucket in every test. Upon accumulating a substantial number of unknown threats/attacks, we apply DBSCAN on unknown instances to make clusters based on similarities. We find the distribution pattern of the largest cluster. We extract the instances of the largest unknown attacks in the cluster. These new instances are added to the original training dataset to retrain both the OCC and RF at the second level. Fig. 2 shows the retraining phases aimed at incorporating a different new category of unknown attack. The sequential approach of retraining the model with different new attack types to enable it to detect them in the future.

- **Initial Setup:** In Fig. 3, the test data is divided into different segments ($Test_1, \ldots, n_1, Test_1, \ldots, n_2, Test_2, \ldots, n_3$). The length of these segments depends on the number of unknown attacks detected by the first-level OCC. Fig. 3 shows three rounds of retraining the model. The process starts with initial training points, indicating that the system has been trained on a dataset with known attack categories.
- **Detection and Classification:** The OCC2 checks incoming unknown attacks from the OCC1. If an attack is recognized as known, it is passed to a multi-classifier (M_1) which determines the specific category of the attack ($a_1, a_2, \ldots, a_n$). If the attack is unknown, it is stored for later processing.
- **Handling Unknown Attacks:** As unknown attack data accumulates, clustering algorithms like DBSCAN are used to group similar attack patterns. This implies that the system has some unsupervised learning capability to handle novel threats. The process focuses on the largest cluster, assessing its distribution to decide if it represents a new type of attack. If confirmed as a new attack type, the data points from this cluster are added to the initial training set.
- **Retraining:** With new data points added, both the OCC2 and M_1 model are retrained to improve their detection and classification abilities. Fig. 3 suggests that multiple rounds of retraining may be necessary to capture all unknown attack categories. Each round incorporates new findings from the largest cluster of unknown attacks at that time.
- **Iterative Process:** The process is iterative, with the potential for several retraining rounds (1st round retraining, 2nd round retraining, 3rd round retraining). This iterative process likely continues until the system no longer encounters new types of unknown attacks, or until all unknown attacks have been classified into new categories and added to the known list.

Our model employs both supervised learning (RF for known attack classification) and unsupervised learning (DBSCAN/DPC for clustering unknown attacks). The system is designed to adapt over time, learning from new types of attacks and expanding its classification abilities through retraining. Our retraining process is presented in Algorithm 1.

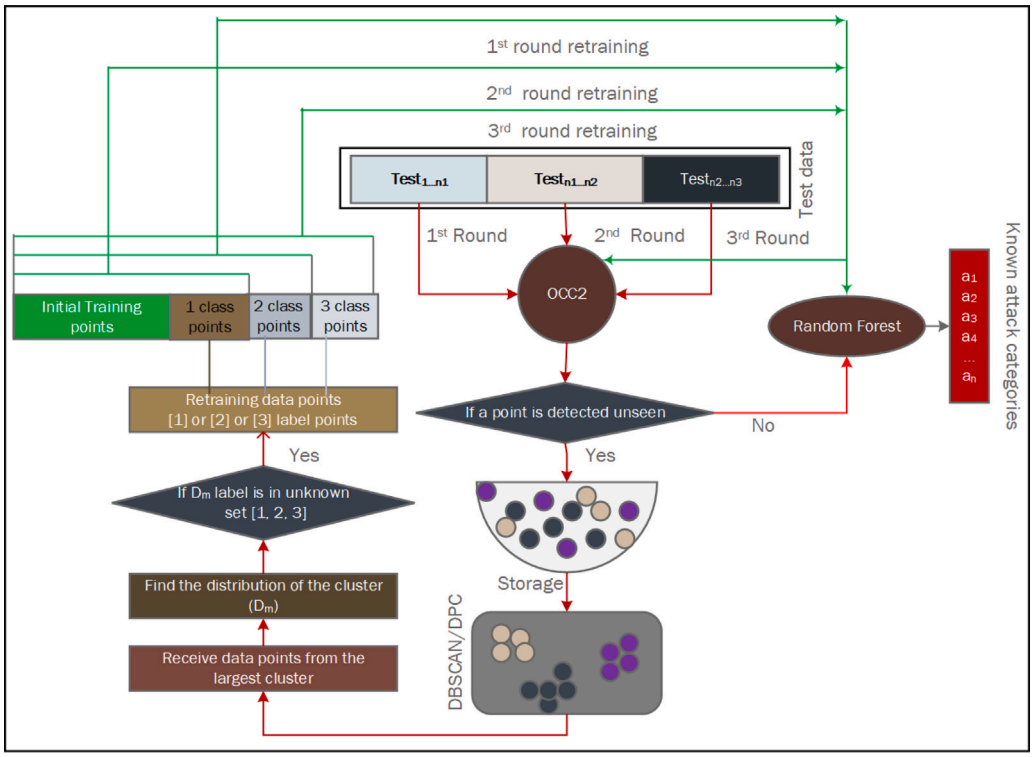


Fig. 2. Simulation of retraining the adaptive IDS.

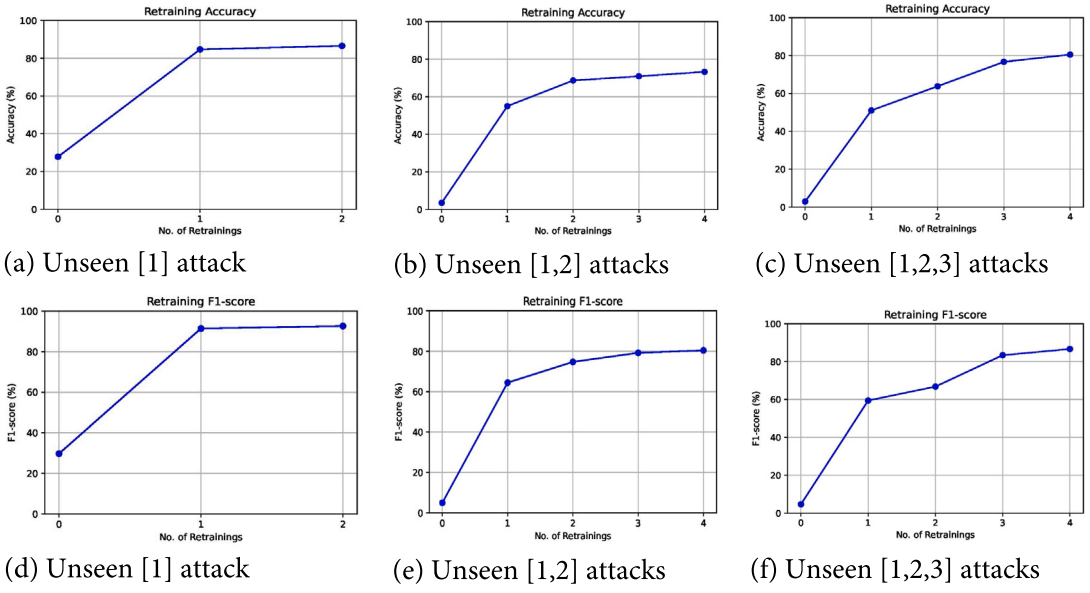


Fig. 3. Impact of retraining on accuracy and F1-score on NSL-KDD datasets.

3.2. Dataset and pre-processing

In this section, we provide a short description of the 10 different IDS benchmark datasets that we have used in this work.

- NSL-KDD dataset [34] was designed to overcome the issues with KDD'99 dataset. This updated version of the KDD data set is still regarded as an effective benchmark dataset for researchers to compare different intrusion detection approaches. The NSL-KDD training and testing sets have a balanced quantity of records for benign and attack samples. The shape of the datasets is (148 517, 44).
- UNSW-NB15 dataset: The Network Security Research Lab at the University of New South Wales, Australia, built the UNSW-NB15 dataset by capturing network traffic in a realistic setting using a high-speed network sniffer and various tools and techniques such as packet flooding, port scanning, and SQL injection. The original dataset contains 257,673 records and 45 fields.
- Canadian Institute for Cybersecurity released CIC-IDS2017 dataset [35] which is a benchmark dataset for Intrusion Detection Systems. The dataset includes user behavior models that are protocol-agnostic through HTTP, HTTPS, FTP, SSH, and email. The dataset consists of 222 914, and 78 features having four classes: benign samples, DoS SlowLoris samples, DoS Slow Httptest

Algorithm 1: Adaptive Attack Detection and Classification

Input : Initial dataset with known attack categories, Test data
Output: Updated detection and classification models

```
# Initial Setup
Train system on the initial dataset with known attack categories;
Divide test data into segments: Test_1, ..., n_1, Test_1, ..., n_2, Test_2, ..., n_3;
Form the different combination of unknown and known attack sets: ([a_1], [a_2, a_3]), ..., ([a_1, a_2..a_{n-1}], [a_n]);
for each unknown-known attack set do
    for each set of test data do
        # Detection and Classification
        for each data point in a testing set do
            Use OCC2 to predict attack class;
            if attack is known then
                Classify attack using multi-classifier M_1;
            end
            else
                Store unknown attack data;
            end
        end
        # Handling Unknown Attacks
        Accumulate data on unknown attacks;
        Apply clustering algorithms to group unknown attacks;
        Focus on the largest cluster;
        if largest cluster represents a new attack type then
            Add data points from cluster to training set;
        end
        # Retraining
        Update training set with new data points;
        Retrain OCC2 and M_1 model;
        if no new unknown attacks are identified then
            break # Exit the loop if no new unknown attacks
            ;
        end
    end
end
```

samples, DoS Hulk samples, DoS GoldenEye samples, and Heartbleed samples in the output class label.

- CIC-DDoS2019: The Canadian Centre for Cybersecurity at the University of New Brunswick created a dataset of DDoS attacks called CIC-DDoS2019. This data set contains both normal traffic patterns and a wide variety of distributed denial of service (DDoS) assaults, such as UDP flood, HTTP flood, and TCP SYN. The shape of the dataset is(431 371, 79) where attack instances are 333 540 and benign instances are 97 831.
- Malmem2022: Obfuscated malware hides them to avoid detection and elimination using conventional anti-malware software. Malmem 2022 [36] is a simulated obfuscated dataset designed to be as realistic as possible to train and test machine learning algorithms to detect obfuscated malware. The dataset is balanced one having level 2 categories: Spyware, Ransomware, and Trojan Horse.
- ToN-IoT-Network and ToN-IoT-Linux: ToN-IoT was extracted from a realistic large-scale IoT simulated environment at the Cyber Range Lab led by ACCS in 2019. The dataset contains heterogeneous telemetry IoT services, traffic flows, and logs of the operating system. Later, Bro-IDS known as Zeek having 44 features was formed from the original dataset considering the network traffic flows. Label encoding is used to convert its categorical features into numerical features following [37,38]. These datasets contain IP addresses. We can treat each unique IP address as a category and perform one-hot encoding. Although this is theoretically possible, it is usually not practical for real-world IDS systems due to the vast number of unique IP addresses, which leads to extremely high-dimensional data.
- ISCXURL2016: In WWW web, URLs serve as the primary mode of transport and attackers insert malware into users' computer systems through URLs. The researchers focus on developing methods for blacklisting malicious URLs. Mamun et al. [39] formed a modern URL dataset that contains the following categories of URLs: benign URLs, spam URLs, phishing URLs, malware URLs and defacement URLs. The shape of the original datasets is (36 707, 80).
- CIC-Darknet2020: The CIC-Darknet dataset has 141 530 records with 85 columns features and was labeled in two ways. We apply label encoding to convert its categorical features into numerical values.
- XIIoTID: The XIIoTID dataset [40] has an initial shape of (596 017, 64). The dataset has features from network traffic, system logs, application logs, device resources (CPU, input/Output, Memory, and others), and commercial Intrusion detection systems' logs (OSSEC and Zeek/Bro).

In this work, we apply label encoding to convert categorical features into numerical features for all of the datasets. We perform the normalization to scale features' values in the range of 0 and 1. Normalization refers to a data pre-processing technique applied in machine learning to standardize the scale of all dataset's attributes. This entails converting the original data to a new range, typically between 0 and 1 or −1 and 1. The min–max normalization formula is as follows:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

where X_norm is the normalized value of X, X_min is the minimum value of X, and X_max is the maximum value of X.

3.3. Experimental setup and implementation

Our experimental study was performed on an Intel Xeon E5-2670 CPU (8 cores, 16 threads), 128 GB DDR3 RAM and 2x Nvidia GTX 1080 Ti. Python 3.9 was used to execute our code. The study utilized ten different machine learning models and primarily relied on Pandas and NumPy libraries for data pre-processing. Since the framework was developed using Python, the widely recognized Scikit-learn toolkit was utilized to leverage its wide range of algorithms and resources for data scientists, including effective accuracy and precision estimation metrics.

3.4. Performance metrics

In this study, we use accuracy, precision, and F1 scores that are essential for assessing the performance of an IDS model. However, their significance can vary depending on the system's specific objectives and requirements. Accuracy quantifies the proportion of accurate classifications made by the IDS. However, relying solely on accuracy is not the most suitable performance metric for IDS, as this might not accurately reflect the system's capability to identify attacks, which are a minority class within the dataset. Precision refers to the proportion of genuine positive detection out of all positive detection. High precision is essential in IDS to minimize false positives, which can result in false alarms. Recall measures the system's ability to reliably identify all instances of a particular class of attack. Low recall suggests that the system is missing some attacks, which can pose a significant security risk.

Table 1
Normal vs. Attack: Root level model's performance.

Datasets	usfAD		LOF		IOF		OCSVM		AE	
	ACC	F1-score	ACC	F1-score	ACC	F1-score	ACC	F1-score	ACC	F1-score
NSL-KDD	**94.90**	**94.90**	69.52	68.15	88.06	88.00	73.07	71.56	89.02	89.00
ToN-IoT-Network	**98.94**	**98.94**	89.31	88.98	63.52	55.32	42.79	43.84	60.76	52.26
CIC-DDoS2019	**98.49**	**97.83**	89.06	85.48	49.30	48.86	82.66	81.79	79.27	80.78
UNSW-NB15	**82.08**	**82.43**	74.65	74.97	55.54	53.25	75.80	74.44	53.96	53.40
Malmem2022	91.48	91.42	88.41	88.39	89.79	89.68	74.97	73.30	**94.93**	**94.92**
ISCXURL2016	**89.57**	**88.35**	80.37	81.89	63.94	67.05	80.85	80.48	73.81	76.16
Darknet2020	90.84	90.39	**93.13**	**93.27**	78.53	75.48	48.61	54.87	76.97	74.64
ToN-IoT-Linux	**97.71**	**97.39**	95.69	95.71	66.87	57.37	42.92	44.29	67.77	67.77
XIIOTID	**93.39**	**93.34**	78.51	76.94	70.60	67.88	68.37	67.42	82.66	82.45
CIC-DDoS2017	**96.98**	**97.10**	83.89	87.37	89.68	90.21	49.72	61.60	86.47	88.26

The F1 score is a combination of precision and recall that quantifies the proportion of true positive identification relative to the total number of positive instances in the dataset. F1-score is a valuable metric for IDS because this considers both false positives and false negatives and provides a balanced score between precision and recall. The accuracy, precision, recall and f1-score are calculated as follows.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \times 100$$

$$\text{Precision} = \frac{TP}{TP+FP} \times 100$$

$$\text{Recall} = \frac{TP}{TP+FN} \times 100$$

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \times 100$$

where TP = true positive, TN = true negative, FP = false positive, and FN = false negative.

We apply 5-fold stratified cross-validation to split datasets for training and testing while preserving the balanced proportion of each class in each fold. The stratified cross-validation provides a more accurate estimate of model performance, particularly when working with imbalanced datasets in which one class has more samples than the other. In this study, we trained and evaluated our adaptive hierarchical models using stratified 5-fold cross-validation. We divided the datasets into five folds of equal size, with each fold containing a proportional representation of the different classes. This process is repeated for all five folds.

4. Results and discussion

In this section, we evaluate the effectiveness of our hierarchical adaptive IDS system. To assess the performance of the proposed model, we use 10 different benchmark IDS datasets that are publicly available. We analyze the performance of the model using usfAD, LOF, IOF, OCSVM and AE semi-supervised learners in terms of average accuracy and f1-score across 10 IDS datasets.

4.1. Performance of semi-supervised learner at the first level

An OCC/semi-supervised model is positioned at the root level in our approach, primarily to separate attack samples from normal network traffic. This model is trained on regular network traffic. In this case, attack instances are not required which can address the limited availability of attack samples for training purposes in real-life situations. In hierarchical architecture, the efficacy of the second-level model is dependent on the performance of the root-level OCC model. In this section, we evaluate various OCC models employed at the first level to differentiate between normal and attack categories across multiple datasets. We focus on key metrics such as accuracy and F1-score, which are standard benchmarks for assessing classification models.

Table 1 indicates that usfAD surpasses other OCC detection methods on the majority of the datasets, including NSL-KDD, ToN-IoT-Network, CIC-DDoS2019, UNSW-NB15, ISCXURL2016, ToN-IoT-Linux, XIIOTID, and CIC-DDoS2017. In the case of the Malmem2022 dataset, the Autoencoder (AE) achieves superior accuracy and F1-score compared to other OCC detection techniques. For the Darknet2020 dataset, the Local OCC Factor (LOF) demonstrates enhanced effectiveness in distinguishing between benign or normal samples and attack instances.

4.2. Performance of semi-supervised learner at the second level

The role of the second-level OCC algorithm is to distinguish between known and unknown attacks (also called zero-day attacks) after the first-level OCC model filtered out the normal instances. The second level model is trained with known attack instances, enabling it to identify unknown attacks based on deviations from these known patterns. In this section, we first evaluate the performance of this second-level OCC model, focusing on its ability to distinguish both known and unknown attacks (in a binary classification). Next, we present its performance which focuses on how well the OCC detects the family type of individual attacks that are labeled as known in the testing samples before and after retraining.

To robustly test the model's capabilities, we have established a pool of unknown and unknown attack categories. This includes combinations of 1-attack, 2-attack, and 3- unknown attack types. We present the results for each of these combinations, demonstrating the model's effectiveness across different scenarios of unknown attacks.

Tables 2 and 3 demonstrate the performance of OCC/semi-supervised models at the second level for binary classification—differentiating between seen and unseen attacks—across various datasets in terms of accuracy and F1-score. The 'C1' and 'C2' columns represent the system's performance for two combinations of varying numbers of unknown attack classes. UA1 represents a scenario with one class of unknown attacks, UA2 denotes the presence of two classes of unknown attacks, and UA3 corresponds to situations involving three classes of unknown attacks.

The usfAD model yields superior results in both accuracy and F1-score for the NSL-KDD, ToN-IoT-Network, CIC-DDoS2019, ToN-IoT-Linux, and XIITOID datasets, outperforming other models like OCSVM, IOF, and AE, which exhibit poorer performance on these datasets. Current research has not yet investigated the ToN-IoT-Network, CIC-DDoS2019, ToN-IoT-Linux, and XIITOID datasets for the detection of known and unknown attacks. This demonstrates the usfAD model's effectiveness in handling the most up-to-date IDS datasets. Nevertheless, with the NSL-KDD dataset, the Autoencoder (AE) achieves relatively improved performance when multiple attack classes are unseen. A comparable pattern is noted for the UNSW-NB15 datasets when employing the IOF model.

Malmem2022 shows a significant challenge with low accuracy and F1-Scores in both C1 and C2, suggesting difficulty in differentiating between seen and unseen attacks. For ISCXURL2016, CIC-DDoS2017, and CIC-Darknet2020 datasets, usfAD and LOF achieves moderately better accuracy and F1-score compared to other datasets.

Table 2
Unknown vs. Known: Second level OCC model's performance in terms of accuracy.

Datasets	No. of UAC	usfAD		LOF		IOF		OCSVM		AE	
		C1	C2	C1	C2	C1	C2	C1	C2	C1	C2
NSL-KDD	UA1	**93.49**	**92.99**	74.77	81.15	74.47	86.29	66.10	41.32	79.26	83.61
	UA2	**94.20**	**93.27**	79.82	75.30	90.06	74.01	67.29	60.13	89.92	78.37
	UA3	**94.66**	**94.14**	71.44	80.02	90.21	90.68	67.62	67.20	89.31	89.82
ToN-IoT-Network	UA1	**96.70**	**96.25**	84.69	87.97	30.12	14.68	6.43	0.66	7.07	1.22
	UA2	**96.91**	**96.92**	86.89	86.28	30.63	33.30	8.73	13.59	8.35	10.63
	UA3	**97.20**	**97.47**	88.42	85.71	33.58	35.10	14.26	13.04	12.14	21.30
UNSW-NB15	UA1	**89.72**	68.42	85.04	64.33	79.97	**85.47**	51.52	45.81	75.41	80.68
	UA2	70.62	85.81	72.33	**87.46**	**80.04**	84.98	55.17	49.54	79.67	78.64
	UA3	64.26	70.22	69.18	76.24	**86.37**	**84.58**	57.28	58.62	81.72	80.56
Malmem2022	UA1	56.60	55.92	54.38	54.13	54.86	52.87	36.87	33.56	**59.69**	**57.63**
	UA2	**47.41**	**42.22**	41.99	37.12	34.74	30.96	40.85	33.40	41.36	34.47
ISCXURL2016	UA1	**78.60**	**80.34**	78.97	74.49	60.64	62.10	43.93	55.26	60.89	59.33
	UA2	**87.15**	**77.05**	73.35	78.75	52.90	47.75	66.41	49.10	47.79	50.91
CIC-DDoS2019	UA1	**89.06**	**95.76**	84.21	83.08	81.33	83.94	17.57	49.08	41.95	88.84
	UA2	**90.63**	**93.00**	79.92	86.88	72.64	87.70	25.50	8.60	40.00	37.32
	UA3	**93.17**	**86.72**	88.00	76.69	91.47	62.48	82.06	25.44	87.10	33.43
CIC-DDoS2017	UA1	**62.80**	**64.06**	40.45	27.10	27.59	13.64	2.38	2.70	34.34	17.68
	UA2	**64.05**	**62.22**	27.07	40.11	17.96	25.58	1.90	2.85	24.87	32.19
	UA3	**64.19**	**65.61**	27.13	29.28	16.94	15.08	4.60	5.56	24.64	25.71
ToN-IoT-Linux	UA1	**84.04**	**82.36**	78.58	76.12	39.65	30.56	10.47	9.02	31.48	32.34
	UA2	**78.51**	**78.76**	77.38	77.46	29.55	30.10	14.80	13.91	30.60	31.71
	UA3	**87.57**	**81.11**	84.75	78.45	29.80	38.56	18.37	18.77	30.81	36.04
XIIOTID	UA1	**94.93**	**94.84**	85.01	92.87	75.12	78.07	62.19	39.52	71.66	79.32
	UA2	**94.91**	**94.90**	85.09	87.15	78.35	76.33	47.16	61.45	74.94	70.43
	UA3	**94.81**	**95.08**	87.23	85.06	78.94	79.17	53.13	50.26	72.48	81.20
CICDarknet2020	UA1	**64.12**	58.62	60.41	**65.06**	16.48	7.21	9.41	3.36	31.63	23.79
	UA2	61.37	67.56	47.28	**62.62**	39.75	16.21	34.36	11.56	44.80	33.43
	UA3	**65.80**	62.31	49.54	39.62	39.77	47.97	36.88	31.51	52.97	44.92

Table 3
Unknown vs. Known: Second level OCC model's performance in terms of F1-score.

Datasets	No. of UAC	usfAD		LOF		IOF		OCSVM		AE	
		C1	C2	C1	C2	C1	C2	C1	C2	C1	C2
NSL-KDD	UA1	**91.33**	**90.26**	73.78	82.66	75.27	87.71	64.70	45.81	79.91	85.12
	UA2	71.30	**90.29**	74.20	74.17	87.13	74.73	58.88	58.10	**87.19**	79.13
	UA3	48.63	70.65	74.03	74.43	**88.80**	86.96	58.14	58.67	86.79	**87.01**
ToN-IoT-Network	UA1	**96.72**	**96.47**	85.01	89.23	15.49	24.62	3.14	0.09	1.66	0.03
	UA2	**96.91**	**96.92**	86.81	86.21	15.87	17.46	6.54	3.36	2.03	2.75
	UA3	**97.18**	**97.46**	88.26	85.57	17.38	19.33	3.73	10.08	3.77	7.50
UNSW-NB15	UA1	**89.61**	49.78	85.04	62.52	76.97	**88.70**	51.50	49.84	68.98	84.70
	UA2	67.48	85.77	74.85	**87.41**	**78.18**	82.40	51.97	47.40	74.18	73.34
	UA3	55.99	59.43	74.64	**79.99**	**84.51**	81.29	54.28	54.76	77.26	74.96
Malmem2022	UA1	47.70	45.65	53.76	**54.63**	54.26	51.96	39.03	36.91	**57.21**	54.01
	UA2	47.28	42.08	41.49	34.88	32.22	26.53	37.49	31.51	38.98	28.54
ISCXURL2016	UA1	**79.75**	**80.41**	79.36	74.42	54.45	59.95	46.17	54.19	54.78	56.45
	UA2	**86.10**	**76.88**	75.10	78.73	56.97	39.36	65.11	48.63	51.04	43.83
CIC-DDoS2019	UA1	89.32	95.79	84.16	84.51	81.34	89.56	17.60	58.64	44.26	92.35
	UA2	91.24	93.17	80.53	86.70	74.49	86.90	28.36	7.62	41.87	43.18
	UA3	93.94	88.30	87.64	78.37	90.81	67.57	77.20	29.83	87.67	37.94
CIC-DDoS2017	UA1	**64.23**	**68.53**	45.88	36.38	42.99	17.98	4.05	1.50	50.72	24.96
	UA2	**61.07**	**61.38**	22.99	44.52	24.07	37.88	2.22	3.69	33.86	46.09
	UA3	**58.12**	**60.83**	21.60	28.20	20.83	19.46	5.65	8.87	31.77	34.56
ToN-IoT-Linux	UA1	**82.27**	**77.77**	78.96	76.64	45.67	41.94	6.40	9.95	37.04	41.14
	UA2	**77.72**	**78.02**	77.36	77.44	32.33	30.30	8.55	7.54	31.63	33.28
	UA3	**88.15**	**81.67**	84.58	78.57	28.36	35.56	10.28	9.18	29.09	32.85
XIIOTID	UA1	**95.02**	**94.93**	84.61	94.09	80.96	76.03	65.56	40.25	72.44	79.75
	UA2	**94.81**	**94.90**	84.79	87.25	76.66	80.33	38.28	63.29	73.92	70.59
	UA3	**94.50**	**94.93**	87.46	85.07	76.38	77.54	43.27	38.72	72.21	78.81
CIC-Darknet2020	UA1	**63.03**	**67.70**	60.35	67.70	23.34	12.04	3.55	4.57	44.60	37.51
	UA2	**58.51**	**64.27**	47.39	62.47	52.41	16.98	42.68	5.51	57.58	44.57
	UA3	60.22	**58.02**	50.31	39.83	49.55	59.12	45.58	38.51	**62.68**	56.95

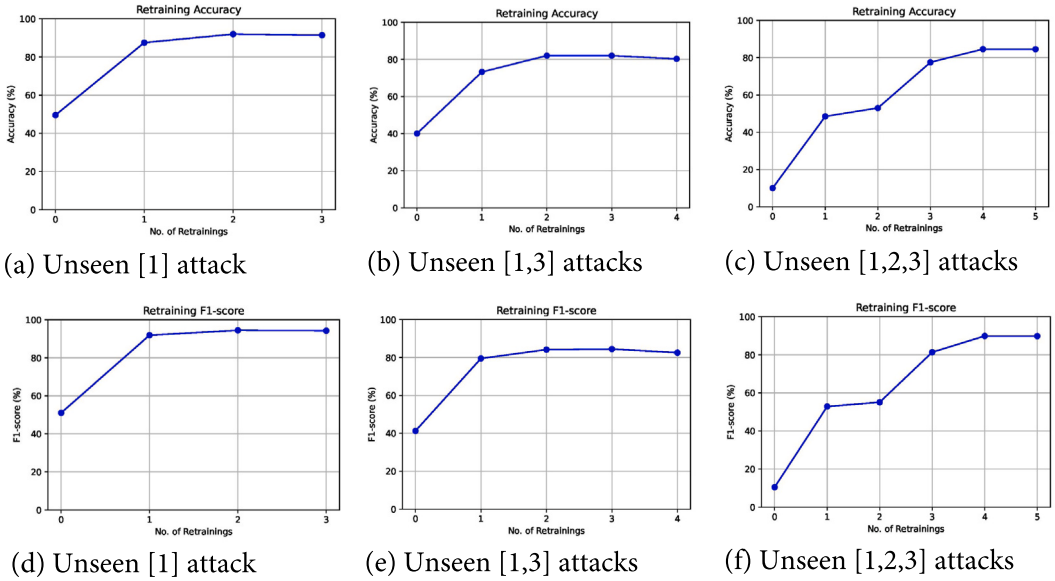


Fig. 4. Impact of retraining on accuracy and F1-score on UNSW-NB15 datasets.

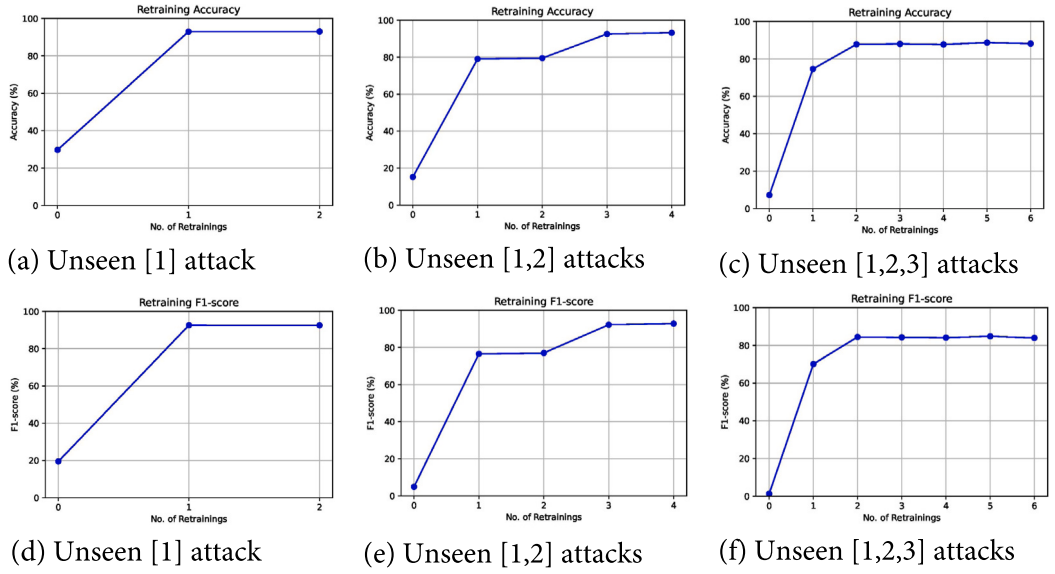


Fig. 5. Impact of retraining on accuracy and F1-score on CICDDoS2019 datasets.

4.3. Performance of semi-supervised learner at the second level in retraining process

We retrain the OCC model at the second level. The model initiates retraining when approximately 1000 unknown points are gathered. These 1000 unknown instances are forwarded to the DBSCAN clustering algorithm, which then forms multiple clusters. We primarily concentrate on the predominant class within the largest cluster. The samples from the dominant class in the largest cluster are added with prior training samples to retrain the second-layer OCC model and the random forest algorithm. On the test dataset, we repeat this process until the clusters incorporate all unknown attack categories for retraining purposes. The necessity for multiple retraining is directly proportional to the quantity of unknown attack samples in the testing set. In this context, we exhibit a retraining progress report that covers scenarios where one, two, and three attack classes are unknown/unseen, utilizing usfAD model (we have chosen usfAD because it shows better results than other methods in detecting known and unknown attack samples across most of the datasets) and four datasets: NSL-KDD, UNSW-NB15, CICDDoS2019, and ToN-IoT-Network, which serve as representatives for other IDS datasets and OCC models. Referencing Figs. 3, 4, 5, and 6, there is a noticeable increasing pattern in both accuracy and F1-score correlating to the varying numbers of unknown attack classes/categories. Herein, we assess the model's performance in learning and detecting various known attacks after retraining.

In this study, we primarily examine how effectively the second-level usfAD and other OCC detection methods learn to identify known attacks after the model retraining is completed. We evaluate the performance of these models in terms of accuracy and average weighted F1-score prior to starting the retraining process in Tables 4 and 5. We observe that the models exhibit slightly higher accuracy and F1-score when only one class is unknown. This is because, in such scenarios, the models have been trained with the majority of attack classes, resulting in somewhat better detection performance for known classes even before retraining begins. However, the performance drops when two or three attack classes are unknown, which is understandable as the

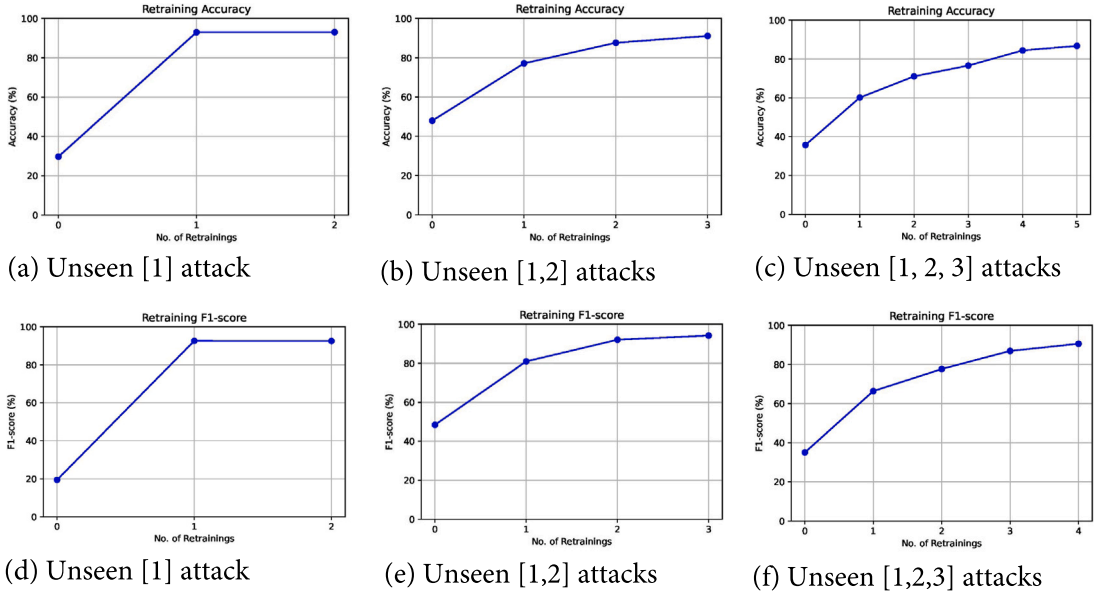


Fig. 6. Impact of retraining on accuracy and F1-score on ToN-IoT-Network datasets.

Table 4
Accuracy of known attack before retraining second level OCC models.

Datasets	No. of UAC	usfAD		LOF		IOF		OCSVM		AE	
		C1	C2	C1	C2	C1	C2	C1	C2	C1	C2
NSL-KDD	UA1	21.88	**72.95**	25.56	59.23	15.44	69.87	24.54	27.36	**26.02**	67.77
	UA2	2.42	19.00	**4.49**	22.65	0.70	14.33	5.74	19.24	3.30	**23.07**
	UA3	0.00	2.32	**19.92**	4.53	1.85	0.34	3.60	**5.60**	2.16	3.17
ToN-IoT-Network	UA1	**60.08**	**84.24**	52.18	73.32	1.08	14.13	1.34	0.04	0.40	0.00
	UA2	**48.24**	**48.07**	39.67	39.46	1.04	0.51	3.06	0.00	0.42	0.40
	UA3	**36.24**	**35.75**	27.26	32.99	0.24	0.68	0.04	4.57	0.67	0.00
UNSW-NB15	UA1	**49.93**	64.66	37.56	59.60	5.12	**80.75**	25.71	31.54	5.64	74.09
	UA2	**19.82**	**40.32**	14.10	30.03	2.39	4.07	14.68	16.06	3.20	4.64
	UA3	**10.49**	**9.34**	7.80	6.75	1.37	1.30	5.41	10.72	1.54	2.07
Malmem2022	UA1	48.93	49.69	48.80	47.45	50.22	50.47	21.94	22.31	**54.21**	**54.56**
	UA2	21.31	23.48	23.11	25.06	24.66	25.58	11.25	10.68	**26.58**	**27.76**
ISCXURL	UA1	56.02	41.47	56.39	42.36	58.21	43.84	27.99	22.78	**59.59**	**43.14**
	UA2	18.81	35.54	16.20	40.94	15.35	42.90	9.35	21.68	**16.60**	**44.34**
CIC-DDoS2019	UA1	**28.87**	82.79	27.15	74.36	14.26	81.79	8.70	40.99	16.47	**86.37**
	UA2	14.91	**21.00**	13.92	19.88	14.39	14.20	7.33	5.49	**17.03**	9.62
	UA3	7.46	10.59	6.87	10.17	14.10	6.35	3.96	5.37	**8.91**	**11.00**
CIC-DDoS2017	UA1	**36.47**	**47.81**	27.71	22.08	27.59	10.78	2.07	0.71	34.33	15.34
	UA2	20.88	27.91	9.71	26.36	15.38	23.85	1.13	1.88	**22.26**	**30.51**
	UA3	11.84	14.92	8.26	13.94	12.61	12.14	2.90	4.72	**20.29**	**22.89**
ToN-IoT-Linux	UA1	**69.04**	**77.95**	60.90	67.30	32.06	28.21	2.70	5.07	23.86	27.43
	UA2	**47.85**	**47.60**	43.22	43.14	19.30	15.93	3.12	2.68	17.36	18.94
	UA3	**25.96**	**32.59**	25.19	26.86	12.58	13.16	3.19	2.25	11.63	12.24
XIIOTID	UA1	63.03	62.25	65.30	**85.02**	**69.24**	21.03	43.74	20.74	49.28	44.69
	UA2	30.46	47.16	**57.94**	39.04	17.24	**64.22**	8.63	41.02	16.79	43.44
	UA3	14.65	23.27	**31.76**	**41.98**	11.92	17.15	6.99	4.95	11.00	10.85
CIC-Darknet2020	UA1	27.18	55.73	**35.68**	**64.07**	13.50	6.44	1.15	2.32	29.93	23.38
	UA2	18.63	19.36	22.07	29.46	38.73	8.71	26.98	1.92	**43.51**	**30.68**
	UA3	10.81	14.79	15.61	18.48	36.26	**46.42**	30.22	23.42	**51.34**	43.28

model gradually learns to recognize these categories through various retraining stages.

After collecting a certain number of unknown attacks (1000 in our simulation), we employ the DBSCAN [41] clustering method to create groups of similar kinds of unknown attacks. Here, we focus on the largest cluster (having the largest members) for retraining our second-level OCC and supervised models. In a real-life scenario, one of the unknown attacks among others might form the majority. In our experiment, we notice that in the initial phase of retraining, the most prevalent unknown attack type forms the largest cluster and is utilized for retraining. In subsequent phases, as these attacks become recognized and classified as known, other types of unknown attacks emerge as the largest cluster. Consequently, our attention is centered on the similarity metrics of the largest cluster.

The effectiveness of this approach depends on the purity of the largest cluster. Therefore, we measure the purity of the largest cluster by dividing the number of members in the dominant class by the total number of members in the cluster. If the number of the dominant class's members is d and the total number of members in the cluster is n. The purity score or the largest cluster ($PSL1$) is calculated as follows:

Table 5
Average weighted F1-score of known attacks before retraining second level OCC models.

Datasets	No. of UAC	usfAD		LOF		IOF		OCSVM		AE	
		C1	C2	C1	C2	C1	C2	C1	C2		
NSL-KDD	UA1	22.68	**73.82**	27.77	63.38	18.32	72.40	**33.56**	37.23	28.18	70.72
	UA2	2.64	19.79	5.48	25.16	1.19	17.21	**9.79**	**27.60**	4.59	25.58
	UA3	0.00	2.57	**20.40**	5.61	3.03	0.52	6.49	**9.58**	3.58	4.43
ToN-IoT-Network	UA1	**60.60**	**85.00**	55.87	76.47	2.07	21.42	2.31	0.08	0.79	0.00
	UA2	**48.77**	**48.54**	42.30	41.49	2.00	1.01	4.89	0.00	0.83	0.79
	UA3	**36.69**	**36.06**	28.24	35.75	0.47	1.31	0.08	6.75	1.33	0.00
UNSW-NB15	UA1	**51.45**	65.76	40.37	65.24	7.63	**83.99**	36.36	41.71	8.92	76.89
	UA2	**20.89**	**41.46**	15.86	32.20	3.85	5.67	21.37	23.80	5.18	7.14
	UA3	**10.90**	10.17	8.63	7.97	1.99	2.26	7.82	**16.23**	2.41	3.47
Malmem2022	UA1	52.15	52.96	52.20	51.82	52.64	53.12	29.23	29.41	**57.15**	**57.45**
	UA2	23.78	25.91	24.89	26.83	25.72	27.29	14.98	14.42	**28.02**	**29.43**
ISCXURL	UA1	60.42	44.04	61.83	46.72	62.70	47.46	40.06	32.19	**64.43**	**46.63**
	UA2	**19.70**	39.20	18.11	44.92	17.04	46.50	13.96	30.65	18.58	**47.98**
CIC-DDoS2019	UA1	29.31	83.36	**28.72**	78.06	15.38	83.22	10.98	51.22	17.86	**88.87**
	UA2	**15.06**	**21.39**	15.33	21.13	14.82	15.42	8.84	8.01	17.21	10.75
	UA3	7.55	10.77	8.08	**11.32**	**14.47**	6.67	4.97	6.01	9.41	11.23
CIC-DDoS2017	UA1	40.76	**51.56**	39.30	26.50	35.09	15.29	2.75	0.98	**47.48**	24.37
	UA2	24.14	31.94	13.64	37.90	21.62	30.09	1.80	2.57	**33.75**	**42.35**
	UA3	14.77	17.53	12.32	20.74	18.14	17.78	5.39	8.54	**31.02**	**34.54**
ToN-IoT-Linux	UA1	**69.25**	**78.14**	63.28	69.97	43.98	40.65	5.22	9.51	30.86	35.67
	UA2	**47.97**	**47.75**	44.99	44.83	27.93	23.89	5.99	5.19	23.17	24.59
	UA3	**26.08**	**32.69**	26.18	28.02	19.83	19.68	6.13	4.36	15.75	14.50
XIIOTID	UA1	**63.85**	**63.21**	67.27	87.60	72.36	23.69	54.67	26.45	53.94	49.10
	UA2	**31.20**	**47.94**	59.88	39.91	18.72	66.69	11.53	51.42	19.05	46.67
	UA3	**15.44**	**23.92**	32.58	43.74	12.66	18.38	9.77	6.83	12.64	12.25
CICDarknet2020	UA1	31.51	61.06	41.20	**69.94**	21.69	10.58	2.23	3.84	**43.72**	35.58
	UA2	22.44	22.29	27.04	34.29	52.90	14.16	41.38	3.74	**58.41**	**44.71**
	UA3	12.99	18.18	19.94	23.35	50.23	60.52	45.02	37.01	**65.07**	**57.95**

$$PSL1 = \frac{d}{n}$$

In addition, we evaluate the effectiveness of DBSCAN in clustering members of the same unknown attack category into the largest group. This is measured by calculating the proportion of the dominant class's members in the cluster (denoted as d) to the total number of ground truth instances (T_n) of that dominant class present in the datasets used for clustering. The formula for this measurement, which we refer to as PSL2, is as follows:

$$PSL2 = \frac{d}{T_n}$$

The purity is between 0 and 1, where 1 indicates that all members of the cluster belong to a single class (perfect purity), and lower values indicate a more mixed cluster. We explain the $PSL1$ and $PSL2$ as follows:

- PSL1 ($\frac{d}{n}$) measures the concentration of the dominant class within the cluster, indicating how homogeneous the cluster is.
- PSL2 ($\frac{d}{T_n}$) measures the coverage of the dominant class within the cluster compared to its overall presence in the dataset, indicating the clustering effectiveness for that particular class.

Using these two metrics, we proposed a new metric for measuring the quality of the largest cluster. We suggest using the harmonic mean of $PSL1$ and $PSL2$ to ensure that both metrics contribute significantly to the result.

$$PSLC = 2 \times \frac{(PSL1 \times PSL2)}{(PSL1 + PSL2)}$$

In addition to DBSCAN, we also evaluated the effectiveness of Density Peak Clustering (DPC) [42] using an implementation from GitHub [43]. Tables 6 and 7 show the comparison of the performance of DBSCAN and DPC algorithms based on three metrics: $PSL1$, $PSL2$, and $PSLC$.

Overall, Table 7 demonstrates that DBSCAN exhibits generally better $PSLC$ scores than DPC, indicating a higher ability to capture the dominant unknown attack categories within the largest cluster. However, Table 6 shows that DPC demonstrates superior performance regarding the purity of the largest cluster, suggesting that its clusters contain a higher proportion of points belonging to the same class. On the other hand, DPC's ability to cover the dominant unknown attack categories within the largest cluster is weaker compared to DBSCAN.

Table 7 provides insights into the strengths and weaknesses of both algorithms for clustering unknown attack categories. The choice between DBSCAN and DPC depends on the relative importance placed on purity and coverage of unknown attack categories within the largest cluster. Considering the overall performance of both algorithms in terms of $PLSC$, $PLS1$ and $PLS2$, we suggested utilizing DBSCAN and analyzed the further performance of the model using the DBSCAN algorithm. Now, we need to analyze the performance of the second-level OCC model and RF after completing retraining.

Next, we present the accuracy and F1-score of known attacks after retraining the second-level OCC models. With retraining the model, it should be able to identify all previously unseen attacks as known. From Tables 8 and 9, we find that the average accuracy and weighted F1-score over 5-fold cross-validation, following the completion of retraining the usfAD model and, in some instances, the LOF, IOF and AE models, are notably high for a majority of the datasets. This includes NSL-KDD, ToN-IoT-Network, UNSW-NB15, CIC-DDoS2019, and XIIOTID.

4.4. Performance of supervised learner at the second level in retraining process

The semi-OCC model, mainly the usfAD algorithm is capable of distinguishing network traffic as either known or unknown. However, for the purpose of assisting security experts in setting appropriate security policies, it is crucial to predict the specific family type of

Table 6
The purity of the largest cluster for some datasets in the retraining process.

Datasets	PLS1						PLS2					
	UA1		UA2		UA3		UA1		UA2		UA3	
	DBSCAN	DPC	DBSCAN	DPC	DBSCAN	DPC	DBSCAN	DPC	DBSCAN	DPC	DBSCAN	DPC
NSL-KDD	1.00	1.00	1.00	1.00	1.00	1.00	**0.69**	0.31	0.69	0.69	0.69	0.69
	1.00	1.00	**0.52**	0.46	**1.00**	0.61	**0.53**	0.44	0.41	**0.47**	0.60	**0.74**
	0.30	**1.00**	0.51	0.51	0.51	**0.99**	0.17	**0.40**	**0.47**	0.38	**0.52**	0.45
CIC-DDoS2019	**0.99**	**0.99**	**0.99**	**0.99**	**0.93**	0.78	**0.97**	0.54	**0.97**	0.49	**0.97**	0.27
	0.94	**1.00**	0.98	**1.00**	0.87	**0.93**	**0.78**	0.53	**0.90**	0.47	**0.99**	0.42
	0.94	**1.00**	0.97	**0.99**	0.92	**1.00**	**0.74**	0.57	**1.00**	0.88	**0.82**	0.46
UNSW-NB15	**1.00**	0.94	**1.00**	**0.91**	**0.98**	0.91	**0.96**	0.48	**0.95**	0.28	**0.95**	0.27
	0.99	0.90	**0.89**	0.89	0.85	**0.89**	**0.74**	0.30	**0.80**	0.29	**0.71**	0.29
	0.49	**0.93**	0.89	0.91	0.88	**0.99**	**0.56**	0.36	**0.71**	0.32	**0.62**	0.43
ToN-IoT-Linux	**0.98**	0.95	0.61	**0.66**	0.38	**0.44**	**0.86**	0.48	**0.86**	0.58	**0.91**	0.54
	0.80	**0.88**	**0.54**	0.48	**0.52**	0.35	**0.43**	0.42	**0.82**	0.39	**0.95**	0.42
	0.46	**0.81**	**0.65**	0.58	0.45	**0.49**	0.24	0.31	**0.51**	0.28	**0.66**	0.39
ToN-IoT-Network	**0.97**	0.96	0.74	**0.81**	0.74	**0.83**	**0.96**	0.27	**0.96**	0.77	**0.95**	0.33
	0.87	**0.88**	0.63	**0.83**	0.70	**0.75**	**0.77**	0.31	**0.97**	0.27	**0.95**	0.93
	0.68	**0.93**	**0.71**	0.63	**1.00**	0.98	**0.57**	0.39	**0.80**	0.27	**0.93**	0.90
XIIOTID	0.99	**1.00**	**1.00**	**1.00**	**0.99**	0.43	**0.65**	0.40	**0.97**	0.89	**0.97**	0.46
	0.99	0.85	**1.00**	0.75	**0.99**	0.61	0.34	**0.51**	**0.65**	0.51	**0.68**	0.41
	0.17	**0.36**	**1.00**	0.88	**1.00**	0.54	0.08	**0.24**	**0.93**	0.50	**0.95**	0.47

Table 7
The comparison between DBSCAN and DPC in terms of the largest cluster's purity.

Datasets	UA1		UA2		UA3	
	DBSCAN	DPC	DBSCAN	DPC	DBSCAN	DPC
NSL-KDD	**0.81**	0.48	0.81	**0.82**	0.81	**0.82**
	0.70	0.61	0.46	0.46	**0.75**	0.67
	0.22	**0.57**	**0.49**	0.43	0.51	**0.61**
CIC-DDoS2019	**0.98**	0.70	**0.98**	0.66	**0.95**	0.40
	0.85	0.70	**0.94**	0.64	**0.93**	0.58
	0.83	0.73	**0.99**	0.94	**0.87**	0.63
UNSW-NB15	**0.98**	0.63	**0.98**	0.43	**0.97**	0.42
	0.84	0.45	**0.84**	0.44	**0.78**	0.44
	0.53	0.52	**0.79**	0.47	**0.73**	0.60
ToN-IoT-Linux	**0.91**	0.64	**0.71**	0.62	**0.53**	0.49
	0.56	**0.57**	**0.65**	0.43	**0.67**	0.38
	0.31	**0.44**	**0.57**	0.38	**0.53**	0.43
ToN-IoT-Network	**0.96**	0.43	**0.83**	0.79	**0.83**	0.47
	0.82	0.46	**0.76**	0.41	0.81	**0.83**
	0.62	0.55	**0.75**	0.38	**0.96**	0.94
XIIOTID	**0.79**	0.58	**0.98**	0.94	**0.98**	0.44
	0.51	**0.64**	**0.79**	0.61	**0.81**	0.49
	0.10	**0.29**	**0.96**	0.64	**0.97**	0.50

network traffic. To do this, we utilize an RF classifier to categorize the known attack types filtered by the second-level OCC model. RF, alongside the semi-OCC model at the second level, is retrained as we accumulate a specific number of unknown attack instances. We need to examine how correctly the RF classifies specific family types of unknown attacks. The F1-scores during the retraining phases of the RF model are depicted in Figs. 7 and 8 for NSL-KDD, UNSW-NB15, CIC-DDoS2019, and ToN-IoT-Network datasets as a representative of other datasets. These graphs illustrate scenarios with varying numbers of unseen/unknown attack categories ([1], [1,2], [1,2,3]). These graphs show that the F1-score of the RF model improves with each retraining cycle. This enhancement indicates the model's growing proficiency in identifying an increased number of unknown attacks after each update with new unknown attack data.

Tables 10 and 11 present the average accuracy and F1-score of the RF classifier across a 5-fold cross-validation, after its retraining with samples of unknown attacks.

4.5. Comparison of our model with a state-of-the-art work

Soltani et al. [27] proposed Deep open classification (DOC++) for identifying benign, known, and unknown attack categories. By analyzing performance, we have recommended the usfAD model developed by Sunil et al. [44] to detect known and unknown attacks in cybersecurity across multiple datasets, against DOC++. In this section, we implemented a two-level hierarchical model by excluding the clustering and retraining process to show the effectiveness of usfAD in detecting known and unknown attacks against the existing DOC++. Here, we exclude the clustering and retraining process because our retraining process differs from the existing works. We design an automatic retraining process focusing on a more practical approach for updating the model while encountering zero-day attacks.

As illustrated in Fig. 9, at the first level, our model distinguishes between benign and attack samples using an OCC method trained solely on benign instances. For the second-level OCC, we divide the training instances into known and unknown categories, using combination $C\binom{n}{k}$ for CIC-IDS2017 and CIC-IDS2018 datasets (Soltani et al. applied these two datasets), where k represents the number of known attacks and (n-k) is the number of unknown attacks. The second level OCC model is trained on known attacks. The second-level model determines if the instances detected as attacks by the first-level model are Known or unknown. Known instances are then classified using an RF (trained on known attacks during the training phase) to specify the family types of the known attacks. Unknown attacks predicted by the second level OCC, in practice, require labeling by a human expert. In this case, we use the ground truth of the unknown attack groups to label their classes. The accuracy of individual attack categories is measured using the following formula, where x indicates a particular attack type.

$$ACC_x = \frac{TP_x}{TP_x + TN_x + FP_x + FN_x}$$

Our model's performance was evaluated both with and without including benign predictions. The average 5-fold accuracy of each attack category showed minimal difference between the two scenarios, for both usfAD and LOF. Tables 12 and 13 compare the accuracy of our model (including usfAD and LOF) with the existing DOC++ model on the CIC-IDS2017 and CIC-IDS2018 datasets. The results demonstrate that usfAD and LOF outperform DOC++ (a deep learning approach) on both datasets.

Table 8
Accuracy of known attacks after updating the OCC models with unknown attacks.

Datasets	No. of UAC	usfAD		LOF		IOF		OCSVM		AE	
		C1	C2	C1	C2	C1	C2	C1	C2	C1	C2
NSL-KDD	A1	**85.87**	**92.74**	56.20	76.07	72.38	89.82	49.91	61.83	87.94	70.23
	A2	**74.41**	**89.73**	55.91	55.42	86.43	84.75	49.37	48.71	74.26	89.32
	A3	**80.26**	**85.91**	44.09	55.73	86.46	87.05	47.42	48.15	80.55	73.33
ToN-IoT-Network	A1	**95.97**	**97.37**	91.67	92.95	34.67	40.38	20.09	22.60	0.67	0.02
	A2	**95.64**	**95.11**	91.44	90.57	35.16	31.75	19.82	11.67	0.91	0.81
	A3	**94.77**	**91.93**	90.18	85.56	30.85	30.39	11.13	16.79	1.20	0.02
UNSW-NB15	A1	76.26	61.89	81.26	66.43	**88.14**	**84.87**	60.58	50.56	8.63	74.52
	A2	64.46	72.44	66.38	79.85	**84.28**	**84.91**	50.55	48.21	57.84	58.68
	A3	59.95	65.68	63.43	74.25	**80.46**	**83.43**	36.56	46.92	69.68	59.57
Malmem2022	A1	49.14	51.47	46.64	51.54	47.00	48.27	38.41	40.13	**81.24**	**65.86**
	A2	37.68	34.20	35.40	31.09	31.03	30.43	30.56	28.82	**52.73**	**50.70**
ISCXURL	A1	78.35	**60.61**	**85.94**	59.74	64.40	48.87	62.89	57.23	59.59	43.14
	A2	**58.30**	63.36	51.05	**67.04**	23.23	47.53	42.23	57.57	49.13	44.34
CIC-DDoS2019	A1	**94.28**	**94.33**	87.78	89.37	80.29	88.56	64.71	40.07	78.40	86.87
	A2	**94.00**	**90.52**	87.48	82.58	88.58	90.06	62.84	74.22	92.94	80.58
	A3	**88.75**	**90.96**	81.51	83.69	88.40	80.68	44.17	62.73	79.22	87.02
CIC-DDoS2017	A1	**70.53**	**64.89**	23.49	20.98	22.91	13.17	5.47	3.57	35.02	22.53
	A2	**56.73**	**61.56**	21.95	22.66	11.23	19.87	3.00	4.40	31.02	31.22
	A3	**48.39**	**49.18**	19.87	18.99	8.74	10.13	2.01	2.61	27.79	30.64
ToN-IoT-Linux	A1	**83.13**	**82.78**	79.69	79.63	37.17	45.25	18.64	19.25	31.55	34.43
	A2	69.78	**75.72**	**70.90**	74.26	31.74	29.66	14.99	15.50	27.91	25.52
	A3	**69.15**	**69.50**	64.54	69.21	23.35	16.06	12.19	11.19	27.64	21.58
XIIOTID	A1	81.61	84.72	**85.80**	**90.80**	71.03	21.51	45.51	18.74	62.19	45.64
	A2	70.07	75.37	**86.57**	**78.48**	34.45	65.26	10.02	43.67	20.93	58.72
	A3	65.86	69.28	**80.79**	**73.80**	54.83	47.10	14.74	5.78	27.33	19.75
CICDarknet2020	A1	**61.86**	**70.80**	57.99	61.55	22.91	23.41	9.62	13.02	33.12	24.28
	A2	**67.65**	**49.77**	57.83	66.02	17.20	18.25	13.13	11.04	45.65	33.10
	A3	**50.29**	**63.71**	44.85	55.55	12.90	16.00	11.08	11.97	51.34	45.99

Table 9
Average weighted F1-score of known attacks after updating the OCC models with unknown attacks.

Datasets	No. of UAC	usfAD		LOF		IOF		OCSVM		AE	
		C1	C2	C1	C2	C1	C2	C1	C2	C1	C2
NSL-KDD	A1	84.58	**90.15**	52.98	68.35	73.01	86.74	45.62	52.16	**93.11**	74.87
	A2	74.90	86.91	51.06	52.62	**86.01**	82.73	46.90	45.03	81.31	**93.11**
	A3	79.73	84.68	45.47	50.95	84.55	86.20	44.88	45.40	86.72	80.82
ToN-IoT-Network	A1	**94.86**	**96.27**	88.75	89.85	31.74	24.81	14.52	8.97	1.32	0.04
	A2	**94.54**	**94.02**	88.59	87.84	26.25	26.38	9.31	5.83	1.78	1.59
	A3	**93.76**	**90.96**	87.60	80.76	23.68	23.90	5.50	7.91	2.35	0.04
UNSW-NB15	A1	72.76	55.86	78.58	60.71	**85.54**	81.40	52.78	40.62	13.95	**77.78**
	A2	62.04	66.26	61.38	76.04	**81.95**	**81.71**	42.13	38.56	68.45	69.38
	A3	57.56	63.90	59.25	73.44	**79.20**	**80.97**	30.91	36.69	74.15	68.49
Malmem2022	A1	40.28	40.17	37.06	39.92	37.63	35.04	28.78	27.29	**86.40**	**69.47**
	A2	26.96	22.67	24.19	17.72	18.38	17.21	18.62	16.94	**56.59**	**54.09**
ISCXURL	A1	72.89	**54.58**	**84.03**	52.46	51.03	32.55	50.82	50.41	64.43	46.63
	A2	55.40	56.79	49.05	**58.69**	14.34	30.63	37.65	48.60	**56.40**	47.98
CIC-DDoS2019	A1	**93.98**	**93.97**	86.10	87.11	80.25	84.89	70.84	50.58	86.25	89.23
	A2	**93.79**	**87.46**	86.38	76.94	86.56	87.62	68.82	78.08	93.44	83.97
	A3	**84.98**	**89.86**	77.29	81.78	85.66	78.28	54.26	67.06	82.67	87.83
CIC-DDoS2017	A1	**59.88**	**55.15**	14.32	13.10	10.16	6.96	1.19	0.77	48.17	32.36
	A2	**45.04**	**49.55**	16.44	13.73	4.99	7.93	0.60	0.67	43.76	43.38
	A3	36.90	37.53	15.06	12.35	4.42	4.85	0.17	0.41	**39.66**	**43.27**
ToN-IoT-Linux	A1	**80.98**	**78.69**	75.86	74.77	27.84	37.08	10.11	8.65	40.99	44.44
	A2	66.03	**74.29**	**66.05**	71.22	23.14	22.80	5.15	5.47	36.45	33.80
	A3	**66.44**	**66.94**	57.15	65.23	15.61	8.17	3.89	3.64	36.33	26.77
XIIOTID	A1	87.30	89.70	**91.14**	**93.85**	74.88	24.21	58.62	24.17	72.02	49.83
	A2	79.67	83.30	**91.23**	**86.15**	45.11	67.58	13.45	56.34	23.83	67.28
	A3	76.32	77.09	**87.40**	**77.37**	62.62	57.84	22.16	7.82	35.74	26.59
CICDarknet2020	A1	**57.32**	**60.68**	52.89	48.46	13.34	12.89	3.13	4.74	47.73	36.70
	A2	57.92	44.76	50.28	**58.50**	7.52	9.61	4.14	3.49	**60.80**	47.60
	A3	44.23	53.19	39.78	47.28	5.09	6.88	4.09	3.56	**65.07**	**61.00**

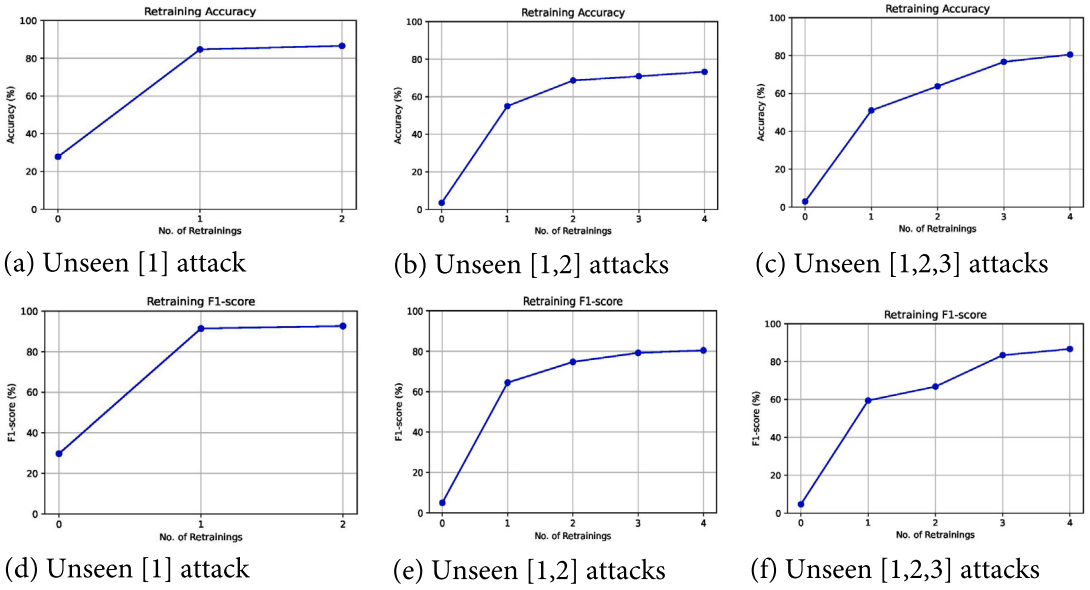


Fig. 7. Impact of retraining on accuracy and F1-score on NSL-KDD and UNSW-NB15.

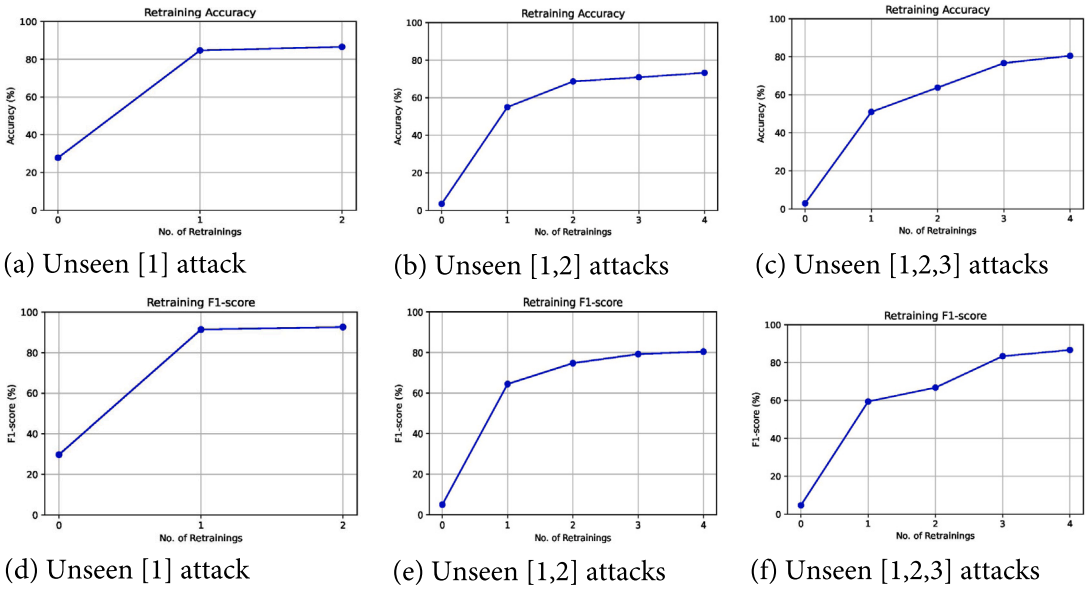


Fig. 8. Impact of retraining on accuracy and F1-score on CIC-DDoS2019 and ToN-IoT-Network.

Furthermore, we present the average 5-fold cross-accuracy of our model for other IDS benchmark datasets in Fig. 10. The usfAD exhibits outstanding performance in detecting known and unknown categories across most datasets, highlighting its potential as a viable option for developing practical IDS models.

4.6. Resolving challenges in dual-tier adaptive ids

In developing an effective Intrusion Detection System (IDS), several challenges must be addressed to ensure the system is robust, scalable, and capable of adapting to new threats. These challenges include the dependence on one-class classifiers, complexity of implementation, scalability, resource intensity, adaptation to new attack vectors, and managing false positives and negatives. The following items outline how each of these issues is identified and the corresponding solutions integrated into our approach to overcome them.

- **Hybrid Methodology and Adaptive Learning:** Our model integrates one-class classification with multi-class classification and clustering, creating a hybrid system that leverages the strengths of each method. This allows for the detection of unknown attacks while managing variations in normal activity, reducing false positives. Adaptive learning techniques automatically adjust to new attack patterns without extensive reconfiguration, further reducing false positives by balancing classifier sensitivity and improving model performance over time.
- **Adaptive Thresholding and Continuous Monitoring:** We implement adaptive thresholding techniques that adjust based on the dynamic nature of normal activity data. These thresholds are set using statistical properties of the training data, such as mean and standard deviation, ensuring a balance between sensitivity and specificity, which helps reduce false positives and false negatives. Continuous monitoring and feedback mechanisms further

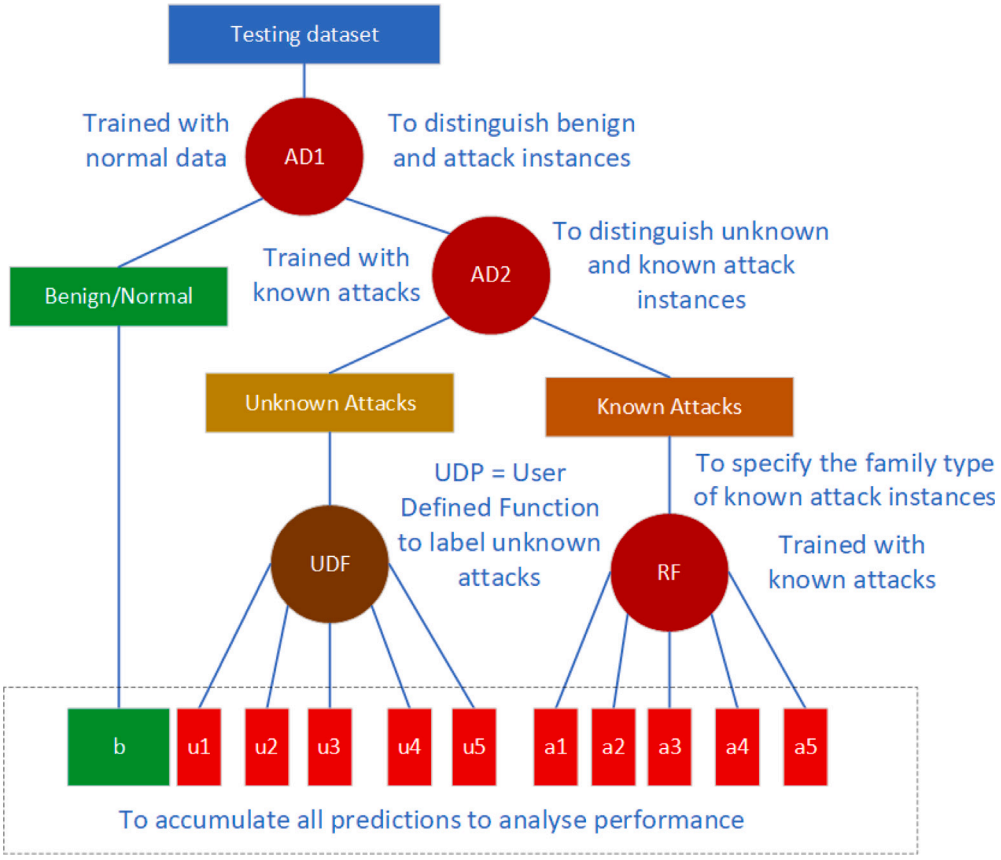
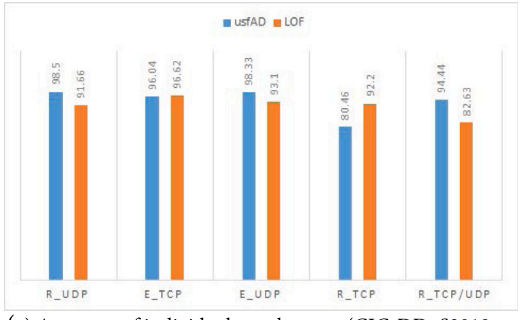
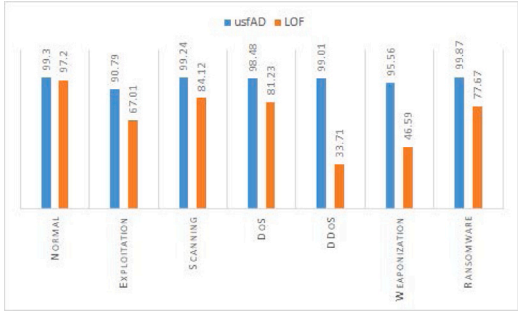


Fig. 9. Hierarchical Simulation of the Proposed Adaptive Framework.

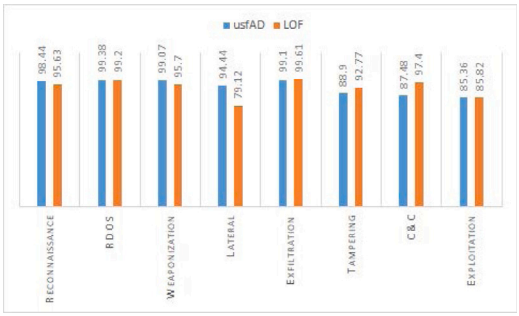


(a) Accuracy of individual attack types (CIC-DDoS2019



(b) Accuracy of individual attack types (ToN-IoT-Network))

(c) Accuracy of individual attack types(UNSW-NB15)



(d) Accuracy of individual attack types(XIIOTID)

Fig. 10. Comparison of accuracy between usfAD and LOF in detecting individual attack families.

Table 10
Accuracy of known attacks after retraining RF's model.

Datasets	No. of UAC	usfAD		LOF		IOF		OCSVM		AE	
		C1	C2	C1	C2	C1	C2	C1	C2	C1	C2
NSL-KDD	UA1	**85.87**	**92.74**	56.20	76.07	72.38	89.82	49.91	61.83	69.81	86.90
	UA2	74.41	**89.73**	55.91	55.42	86.43	84.75	49.37	48.71	**83.55**	67.40
	UA3	80.26	**85.91**	44.09	55.73	86.46	87.05	47.42	48.15	**81.93**	83.37
ToN-IoT-Network	UA1	**95.97**	**97.37**	91.67	92.95	34.67	40.38	20.09	22.60	24.00	24.79
	UA2	**95.64**	**95.11**	91.44	90.57	35.16	31.75	19.82	11.67	23.70	21.60
	UA3	**94.77**	**91.93**	90.18	85.56	30.85	30.39	11.13	16.79	20.43	11.30
UNSW-NB15	UA1	76.26	61.89	81.26	66.43	**88.14**	**84.87**	60.58	50.56	83.08	76.81
	UA2	64.46	72.44	66.38	79.85	**84.28**	**84.91**	50.55	48.21	76.09	77.92
	UA3	59.95	65.68	63.43	74.25	**80.46**	**83.43**	36.56	46.92	71.05	74.29
Malmem2022	UA1	49.14	51.47	46.64	51.54	47.00	48.27	38.41	40.13	**50.58**	**53.16**
	UA2	37.68	34.20	35.40	31.09	31.03	30.43	30.56	28.82	**35.57**	**32.17**
ISCXURL	UA1	**78.35**	**60.61**	85.94	59.74	64.40	48.87	62.89	57.23	65.96	47.40
	UA2	**58.30**	**63.36**	51.05	67.04	23.23	47.53	42.23	57.57	29.37	48.77
CIC-DDoS2019	UA1	**94.28**	**94.33**	87.78	89.37	80.29	88.56	31.56	75.37	45.94	89.89
	UA2	**94.00**	**90.52**	87.48	82.58	88.58	90.06	26.06	16.85	43.71	65.12
	UA3	**88.75**	**90.96**	81.51	83.69	88.40	80.68	71.22	24.44	84.15	54.82
CIC-DDoS2017	UA1	**70.53**	**64.89**	23.49	20.98	22.91	13.17	5.47	3.57	17.76	10.17
	UA2	**56.73**	**61.56**	21.95	22.66	11.23	19.87	3.00	4.40	10.23	16.11
	UA3	**48.39**	**49.18**	19.87	18.99	8.74	10.13	2.01	2.61	7.46	9.26
ToN-IoT-Linux	UA1	**83.13**	**82.78**	79.69	79.63	37.17	45.25	18.64	19.25	43.55	47.59
	UA2	**69.78**	**75.72**	70.90	74.26	31.74	29.66	14.99	15.50	33.91	33.86
	UA3	**69.15**	**69.50**	64.54	69.21	23.35	16.06	12.19	11.19	26.35	29.53
XIIOTID	UA1	**90.66**	**96.80**	83.56	96.22	80.72	82.26	51.00	57.76	77.77	84.80
	UA2	**88.50**	**85.13**	88.03	78.55	76.13	81.36	41.88	45.11	52.34	75.46
	UA3	**85.22**	**88.51**	84.05	76.44	70.85	75.66	30.45	40.30	43.49	46.28
CIC-Darknet2020	UA1	**61.86**	**70.80**	57.99	61.55	57.99	61.55	9.62	13.02	17.68	20.10
	UA2	**67.65**	**49.77**	57.83	66.02	57.83	66.02	13.13	11.04	17.25	14.88
	UA3	**50.29**	**63.71**	44.85	55.55	44.85	55.55	11.08	11.97	11.46	16.36

Table 11
Average weighted F1-score of known attacks after retraining RF's model.

Datasets	No. of UAC	usfAD		LOF		IOF		OCSVM		AE	
		C1	C2	C1	C2	C1	C2	C1	C2	C1	C2
NSL-KDD	UA1	**84.58**	**90.15**	52.98	68.35	73.01	86.74	45.62	52.16	69.44	83.18
	UA2	74.90	**86.91**	51.06	52.62	**86.01**	82.73	46.90	45.03	81.43	67.41
	UA3	79.73	84.68	45.47	50.95	**84.55**	**86.20**	44.88	45.40	79.55	81.31
ToN-IoT-Network	UA1	**94.86**	**96.27**	88.75	89.85	31.74	24.81	20.09	14.52	16.03	12.49
	UA2	**94.54**	**94.02**	88.59	87.84	26.25	26.38	19.82	9.31	16.26	14.97
	UA3	**93.76**	**90.96**	87.60	80.76	23.68	23.90	11.13	5.50	12.18	7.85
UNSW-NB15	UA1	72.76	55.86	78.58	60.71	**85.54**	**81.40**	52.78	40.62	78.98	73.07
	UA2	62.04	66.26	61.38	76.04	**81.95**	**81.71**	42.13	38.56	73.18	73.74
	UA3	57.56	63.90	59.25	73.44	**79.20**	**80.97**	30.91	36.69	69.77	70.86
Malmem2022	UA1	40.28	40.17	37.06	39.92	37.63	35.04	28.78	27.29	**41.40**	**40.78**
	UA2	**26.96**	**22.67**	24.19	17.72	18.38	17.21	18.62	16.94	24.00	18.29
ISCXURL	UA1	**72.89**	**54.58**	84.03	52.46	51.03	32.55	50.82	50.41	53.09	30.68
	UA2	**55.40**	**56.79**	49.05	58.69	14.34	30.63	37.65	48.60	24.20	31.97
CIC-DDoS2019	UA1	**93.98**	**93.97**	86.10	87.11	80.25	84.89	28.35	68.86	49.50	87.81
	UA2	**93.79**	**87.46**	86.38	76.94	86.56	87.62	26.79	10.25	47.28	65.82
	UA3	**84.98**	**89.86**	77.29	81.78	85.66	78.28	66.36	26.01	79.41	57.30
CIC-DDoS2017	UA1	**59.88**	**55.15**	14.32	13.10	10.16	6.96	1.19	0.77	6.44	4.27
	UA2	**45.04**	**49.55**	16.44	13.73	4.99	7.93	0.60	0.67	4.44	5.49
	UA3	**36.90**	**37.53**	15.06	12.35	4.42	4.85	0.17	0.41	3.34	3.87
ToN-IoT-Linux	UA1	**80.98**	**78.69**	75.86	74.77	27.84	37.08	10.11	8.65	31.48	35.48
	UA2	**66.03**	**74.29**	66.05	71.22	23.14	22.80	5.15	5.47	21.13	20.88
	UA3	**66.44**	**66.94**	57.15	65.23	15.61	8.17	3.89	3.64	17.23	16.09
XIIOTID	UA1	**89.55**	**95.34**	82.85	94.46	78.13	78.14	44.26	46.94	74.84	78.70
	UA2	**88.16**	**84.16**	86.44	76.27	73.40	79.12	36.82	38.95	45.77	72.54
	UA3	**83.88**	**85.98**	81.90	71.10	68.21	72.69	24.61	36.81	37.80	43.30
CIC-Darknet2020	UA1	**57.32**	**60.68**	52.89	48.46	52.89	48.46	3.13	4.74	8.77	10.50
	UA2	**57.92**	**44.76**	50.28	58.50	50.28	58.50	4.14	3.49	8.45	7.24
	UA3	**44.23**	**53.19**	39.78	47.28	39.78	47.28	4.09	3.56	4.34	7.64

enhance accuracy by refining the model in real-time and ensuring up-to-date detection capabilities.

- **Incremental Learning, Real-Time Clustering, and Batch Processing:** The system supports incremental learning and real-time

Table 12
Comparison of accuracy between the proposed model and an existing model on CIC-IDS2017.

Attack	usfAD	usfAD (NN)	DOC++	DOC++ (NN)	LOF	LOF (NN)
Benign	98.19	N/A	50.2	N/A	85.05	N/A
DoS Hulk	97.74	97.72	49.01	47.08	90.32	90.35
DDoS	98.58	98.91	47.64	56.56	96.12	96.4
PortScan	98.46	99.38	87.61	85.55	96.79	97.09
DoS GoldenEye	97.36	97.41	54.76	50.26	98.15	98.15
FTP-Patator	99.75	99.72	50.04	61.92	80.05	92.12
DoS slowloris	93.06	93.49	60.16	54.46	97.31	97.97
DoS Slowhttptest	96.28	94.57	63.09	69.13	96.51	96.45
SSH-Patator	95.11	98.93	51.34	52.19	69.44	91.36
Botnet	98.69	98.47	63.14	71.82	98.38	98.07
Web Brute Force	90.14	62.72	40.44	42.24	81.09	79.28
Web XSS	90	50.18	37.33	38.92	86.47	78.9
Infiltration	90	90	N/A	N/A	90.91	91.94
Web Sql Injection	90	79.52	N/A	N/A	80	83.81
Heartbleed	100	100	N/A	N/A	100	100

Table 13
Comparison of accuracy between the proposed model and an existing model on CIC-IDS2018.

Benign/Attack categories	usfAD	LOF	DOC++ (Closed set)	DOC++ (Open set)
Benign	97.5	93.71	88.05	48.79
DDOS attack-HOIC	98.8	97.91	N/A	N/A
DDoS attacks-LOIC-HTTP	98.95	98.89	90.04	72.16
DoS attacks-Hulk	95.92	98.95	N/A	N/A
Botnet	94.89	99.66	92.03	53
Infilteration	97.45	97.17	85.89	48.43
SSH-Bruteforce	99.36	99.68	92.37	42.45
FTP-BruteForce	97.99	99.89	92.41	41.13
DoS attacks-SlowHTTPTest	99.86	99.9	N/A	N/A
DoS attacks-GoldenEye	94.93	98.79	93.17	54.74
DoS attacks-Slowloris	90.26	98.76	92.17	57.26
DDOS attack-LOIC-UDP	90.06	32.66	N/A	N/A
Brute Force -Web	89.18	91.15	88.93	40.06
Brute Force-XSS	83.91	89.13	N/A	N/A
SQL Injection	95.56	100	82.34	35.47

clustering using algorithms like DBSCAN and DPC. This enables the model to update and retrain on new attack data without needing to reprocess the entire dataset, improving adaptability to evolving threats. Batch processing allows the model to handle new data in manageable chunks, reducing computational load. Efficient clustering identifies attack patterns without predefined cluster numbers, ensuring scalability and real-time responsiveness.

- **Dual-Tier Architecture and Clustering for Unknown Attacks:** The two-tier structure of our IDS refines classification by separating the detection of normal versus attack activities in the first tier and distinguishing known versus unknown attacks in the second tier. This hierarchical approach reduces false positives and negatives. Clustering algorithms help group similar unknown attacks, refining the detection process by analyzing their characteristics and reducing the chances of false negatives.
- **Unified Framework, Streamlined Processes, and Distributed Architecture:** Our model integrates one-class classification, multi-class classification, and clustering into a unified framework, simplifying deployment and maintenance. The streamlined design reduces the complexity of managing multiple systems and automates updates and optimizations, making implementation easier in real-world environments. Additionally, the distributed architecture leverages multiple processing units, enabling real-time analysis and enhancing scalability with increased data volumes.
- **Resource Allocation, Adaptive Learning Rate, and Model Pruning:** We employ resource-efficient clustering algorithms optimized for performance, ensuring minimal resource consumption. The system allows flexible resource allocation across multiple processors or servers, balancing computational tasks. An adaptive learning rate mechanism adjusts the retraining frequency based on new data volume and resource availability, while periodic model pruning maintains efficiency, ensuring responsiveness as the dataset grows.
- **Automated Feedback Loop and Continuous Monitoring:** Our system includes an automated feedback loop that collects and analyzes new attack instances, allowing for rapid adaptation to new threats. Continuous monitoring of network traffic and attack patterns ensures prompt detection and classification of new attacks, dynamically adjusting thresholds and model parameters to maintain effectiveness.

These combined strategies ensure our approach effectively addresses challenges related to dependence on one-class classifiers, complexity of implementation, scalability, resource intensity, adaptation to new attack vectors, and the balance between false positives and negatives. By integrating adaptive learning, efficient clustering, incremental updates, and real-time monitoring, our system remains practical, scalable, and robust in dynamic, high-throughput environments.

5. Conclusion

In our work, we propose a novel two-stage framework that integrates both binary and multi-classification models. The first stage filters out benign samples using binary classification, reducing the number of samples that proceed to the second stage. The second stage then classifies the remaining malicious samples into unknown and known instances where a supervised learner classifies known attacks into their respective attack categories. At this level, we employ a DBSCAN which clusters together similar types of unknown attacks for retraining the second-level OCC model and supervised learner. Our research reveals that usfAD, a sophisticated OCC detection model developed recently, outperforms other state-of-the-art algorithms in identifying both known and novel attack patterns across most datasets. The efficacy of usfAD is significantly influenced by the accuracy of its initial model, which is designed to distinguish between normal and potential attacks. The retraining phase's effectiveness depends on the precision with which the clustering mechanism can categorize various new attack types. Through our analysis, we have determined that current implementations of the DBSCAN algorithm outshine alternative clustering methods.

In our study, we proposed a novel dual-tier adaptive one-class classification IDS that improves upon existing methods by integrating adaptive learning mechanisms to address emerging cyber threats. Unlike previous approaches, which often rely on static models or limited datasets, our method dynamically adjusts to new threats and utilizes more comprehensive datasets. This enhancement provides superior adaptability and accuracy compared to traditional IDS models. Overall, while earlier studies have laid the groundwork for IDS development, our approach offers a more advanced and flexible solution tailored to the evolving nature of cyber threats.

For future studies, we propose exploring the development of attention-based IDS models to enhance feature relevance and improve detection accuracy. Additionally, investigating feature fusion techniques to combine multiple data representations could lead to more robust detection. A hybrid clustering approach integrating deep learning models with clustering algorithms could also be explored to further advance IDS capabilities.

CRediT authorship contribution statement

Md. Ashraf Uddin: Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Data curation. **Sunil Aryal:** Conceptualization, Writing – review & editing, Validation, Supervision, Resources, Project administration, Investigation, Funding acquisition, Formal analysis. **Mohamed Reda Bouadjenek:** Visualization, Validation, Investigation, Formal analysis. **Muna Al-Hawawreh:** Visualization, Validation, Investigation, Formal analysis. **Md. Alamin Talukder:** Validation, Resources, Methodology, Investigation, Data curation, Writing – review & editing.

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.

Acknowledgment

This material is based upon work supported by the Air Force Office of Scientific Research under award number FA2386-23-1-4003.

Data availability

The datasets used in this study are retrieved from freely available and open-access repositories, accessible at: https://www.unb.ca/cic/datasets/index.html, https://research.unsw.edu.au/projects/unsw-nb15-dataset.

References

[1] M.A. Talukder, S. Sharmin, M.A. Uddin, M.M. Islam, S. Aryal, MLSTL-WSN: machine learning-based intrusion detection using SMOTETomek in WSNs, Int. J. Inf. Secur. 23 (3) (2024) 2139–2158.
[2] W.F. Urmi, M.N. Uddin, M.A. Uddin, M.A. Talukder, M.R. Hasan, S. Paul, M. Chanda, J. Ayoade, A. Khraisat, R. Hossen, et al., A stacked ensemble approach to detect cyber attacks based on feature selection techniques, Int. J. Cogn. Comput. Eng. 5 (2024) 316–331.
[3] S. Roshan, Y. Miche, A. Akusok, A. Lendasse, Adaptive and online network intrusion detection system using clustering and extreme learning machines, J. Franklin Inst. 355 (4) (2018) 1752–1779.
[4] G. Folino, P. Sabatino, Ensemble based collaborative and distributed intrusion detection systems: A survey, J. Netw. Comput. Appl. 66 (2016) 1–16.
[5] N. Hubballi, V. Suryanarayanan, False alarm minimization techniques in signature-based intrusion detection systems: A survey, Comput. Commun. 49 (2014) 1–17.
[6] L. Bilge, T. Dumitraş, Before we knew it: an empirical study of zero-day attacks in the real world, in: Proceedings of the 2012 ACM Conference on Computer and Communications Security, 2012, pp. 833–844.
[7] P. Joshi, N. Jain, G. Ramtekkar, G.S. Virdi, Vibration and buckling analysis of partially cracked thin orthotropic rectangular plates in thermal environment, Thin-Walled Struct. 109 (2016) 143–158.
[8] Y. Yang, S. Zhu, G. Cao, Improving sensor network immunity under worm attacks: A software diversity approach, Ad Hoc Netw. 47 (2016) 26–40.
[9] M. Hossain, S.M. Bridges, R.B. Vaughn, Adaptive intrusion detection with data mining, in: SMC'03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme-System Security and Assurance (Cat. No. 03CH37483), Vol. 4, IEEE, 2003, pp. 3097–3103.
[10] M. Masdari, H. Khezri, A survey and taxonomy of the fuzzy signature-based intrusion detection systems, Appl. Soft Comput. 92 (2020) 106301.
[11] J.F.C. Joseph, A. Das, B.-S. Lee, B.-C. Seet, CARRADS: Cross layer based adaptive real-time routing attack detection system for MANETS, Comput. Netw. 54 (7) (2010) 1126–1141.
[12] M.A. Talukder, R. Hossen, M.A. Uddin, M.N. Uddin, U.K. Acharjee, Securing transactions: A hybrid dependable ensemble machine learning model using iht-lr and grid search, Cybersecurity 7 (1) (2024) 32.
[13] M.A. Talukder, M.M. Islam, M.A. Uddin, M. Kazi, M. Khalid, A. Akhter, M. Ali Moni, Toward reliable diabetes prediction: Innovations in data engineering and machine learning applications, Digit. Health 10 (2024) 20552076241271867.
[14] M.A. Talukder, M.M. Islam, M.A. Uddin, A. Akhter, K.F. Hasan, M.A. Moni, Machine learning-based lung and colon cancer detection using deep feature extraction and ensemble learning, Expert Syst. Appl. 205 (2022) 117695.
[15] M.A. Talukder, M.M. Islam, M.A. Uddin, K.F. Hasan, S. Sharmin, S.A. Alyami, M.A. Moni, Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction, J. Big Data 11 (1) (2024) 33.
[16] M.A. Talukder, K.F. Hasan, M.M. Islam, M.A. Uddin, A. Akhter, M.A. Yousuf, F. Alharbi, M.A. Moni, A dependable hybrid machine learning model for network intrusion detection, J. Inf. Secur. Appl. 72 (2023) 103405.
[17] V.H. Bezerra, V.G.T. da Costa, S. Barbon Junior, R.S. Miani, B.B. Zarpelão, IoTDS: A one-class classification approach to detect botnets in internet of things devices, Sensors 19 (14) (2019) 3188.
[18] U.M. Fahad, S. Muhammad, Y. Bi, Applying one-class classification techniques to IP flow records for intrusion detection, Balt. J. Mod. Comput. 5 (1) (2017) 70–86.
[19] N. Anand, M. Saifulla, An efficient IDS for slow rate HTTP/2.0 DoS attacks using one class classification, in: 2023 IEEE 8th International Conference for Convergence in Technology (I2CT), IEEE, 2023, pp. 1–9.
[20] P. Dini, A. Begni, S. Ciavarella, E. De Paoli, G. Fiorelli, C. Silvestro, S. Saponara, Design and testing novel one-class classifier based on polynomial interpolation with application to networking security, IEEE Access 10 (2022) 67910–67924.
[21] W.L. Al-Yaseen, Z.A. Othman, M.Z.A. Nazri, Real-time multi-agent system for an adaptive intrusion detection system, Pattern Recognit. Lett. 85 (2017) 56–64.
[22] U.K. Singh, C. Joshi, D. Kanellopoulos, A framework for zero-day vulnerabilities detection and prioritization, J. Inf. Secur. Appl. 46 (2019) 164–172.
[23] M. Al-Zewairi, S. Almajali, M. Ayyash, Unknown security attack detection using shallow and deep ANN classifiers, Electronics 9 (12) (2020) 2006.
[24] H. Hindy, R. Atkinson, C. Tachtatzis, J.-N. Colin, E. Bayne, X. Bellekens, Utilising deep learning techniques for effective zero-day attack detection, Electronics 9 (10) (2020) 1684.
[25] M.A. Talukder, M. Khalid, M.A. Uddin, An integrated multistage ensemble machine learning model for fraudulent transaction detection, Journal of Big Data (2024) http://dx.doi.org/10.1186/s40537-024-00996-5.
[26] M.A. Uddin, S. Aryal, M.R. Bouadjenek, M. Al-Hawawreh, M.A. Talukder, usfAD based effective unknown attack detection focused IDS framework, Scientific Reports (2024).
[27] M. Soltani, B. Ousat, M.J. Siavoshani, A.H. Jahangir, An adaptable deep learning-based intrusion detection system to zero-day attacks, J. Inf. Secur. Appl. 76 (2023) 103516.
[28] K. Sethi, R. Kumar, D. Mohanty, P. Bera, Robust adaptive cloud intrusion detection system using advanced deep reinforcement learning, in: Security, Privacy, and Applied Cryptography Engineering: 10th International Conference, SPACE 2020, Kolkata, India, December 17–21, 2020, Proceedings 10, Springer, 2020, pp. 66–85.
[29] X. Gao, C. Shan, C. Hu, Z. Niu, Z. Liu, An adaptive ensemble machine learning model for intrusion detection, Ieee Access 7 (2019) 82512–82521.
[30] M. Nkongolo, J.P. Van Deventer, S.M. Kasongo, S.R. Zahra, J. Kipongo, A cloud based optimization method for zero-day threats detection using genetic algorithm and ensemble learning, Electronics 11 (11) (2022) 1749.
[31] S. Ali, S.U. Rehman, A. Imran, G. Adeem, Z. Iqbal, K.-I. Kim, Comparative evaluation of AI-based techniques for zero-day attacks detection, Electronics 11 (23) (2022) 3934.
[32] A.E. Topcu, Y.I. Alzoubi, E. Elbasi, E. Camalan, Social media zero-day attack detection using TensorFlow, Electronics 12 (17) (2023) 3554.
[33] N.A. Elfeshawy, O.S. Faragallah, Divided two-part adaptive intrusion detection system, Wirel. Netw. 19 (2013) 301–321.
[34] T. Su, H. Sun, J. Zhu, S. Wang, Y. Li, BAT: Deep learning methods on network intrusion detection using NSL-KDD dataset, IEEE Access 8 (2020) 29575–29585.
[35] H.H. Jazi, H. Gonzalez, N. Stakhanova, A.A. Ghorbani, Detecting HTTP-based application layer DoS attacks on web servers in the presence of sampling, Comput. Netw. 121 (2017) 25–36.
[36] T. Carrier, P. Victor, A. Tekeoglu, A.H. Lashkari, Detecting obfuscated malware using memory feature engineering., in: ICISSP, 2022, pp. 177–188.
[37] N. Moustafa, A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets, Sustainable Cities Soc. 72 (2021) 102994.
[38] G. Guo, X. Pan, H. Liu, F. Li, L. Pei, K. Hu, An IoT intrusion detection system based on TON IoT network dataset, in: 2023 IEEE 13th Annual Computing and Communication Workshop and Conference, CCWC, IEEE, 2023, pp. 0333–0338.
[39] M.S.I. Mamun, M.A. Rathore, A.H. Lashkari, N. Stakhanova, A.A. Ghorbani, Detecting malicious urls using lexical analysis, in: Network and System Security: 10th International Conference, NSS 2016, Taipei, Taiwan, September 28-30, 2016, Proceedings 10, Springer, 2016, pp. 467–482.
[40] M. Al-Hawawreh, E. Sitnikova, N. Aboutorab, X-IIoTID: A connectivity-agnostic and device-agnostic intrusion data set for industrial internet of things, IEEE Internet Things J. 9 (5) (2022) 3962–3977, http://dx.doi.org/10.1109/JIOT.2021.3102056.
[41] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al., A density-based algorithm for discovering clusters in large spatial databases with noise, in: Kdd, Vol. 96, 1996, pp. 226–231.
[42] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, Sci. 344 (6191) (2014) 1492–1496.
[43] Shane, Density peak clustering, 2023, URL https://github.com/freesinger/density_peak_clustering.
[44] S. Aryal, K. Santosh, R. Dazeley, usfAD: a robust anomaly detector based on unsupervised stochastic forest, Int. J. Mach. Learn. Cybern. 12 (2021) 1137–1150.