

RESEARCH ARTICLE

An Incremental Majority Voting Approach for Intrusion Detection System Based on Machine Learning

ALIMOV ABDULBORIY¹ AND JI SUN SHIN², (Member, IEEE)

¹Department of Computer and Information Security, and Convergence Engineering for Intelligent Drone, Sejong University, Seoul 05006, South Korea

²Department of Computer and Information Security, Sejong University, Seoul 05006, South Korea

Corresponding author: Ji Sun Shin (js shin@sejong.ac.kr)


This work was supported in part by the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant 2020R1A6A1A03038540, and in part by NRF funded by the Ministry of Science and ICT under Grant 2020R1F1A1072275.

ABSTRACT With the rapid growth of digitalization and the increasing volume of data, the cybersecurity threat landscape is expanding at an alarming rate. Intrusion Detection Systems (IDS) have been widely employed in conjunction with firewalls to safeguard networks. However, traditional IDS systems operate in a static manner, rendering them vulnerable to obsolescence and necessitating costly retraining efforts. As a result, the demand for dynamic models capable of handling continuous streams of network traffic has surged as they can learn from the incoming traffic without the need to old data and costly retraining. In response to this challenge, we have implemented an enhanced approach: an incremental majority voting IDS system, which utilizes existing tools and techniques to improve the robustness and adaptability of intrusion detection. By leveraging the collective decision-making power of multiple machine learning models such as: KNN classifier, Softmax Regressor and Adaptive Random Forest classifier, our system aims to improve the accuracy, especially reducing false alarm rates, and effectiveness of intrusion detection in real-time scenarios. Through this research, we have managed to obtain a model which scores 96.43% of accuracy as well as giving 100% precision for majority type of attacks. By successfully handling the imbalanced nature of streaming data, our adopted model shows promising potential as a high-performing solution for IDS and can be considered one of the robust IDS models capable of dealing with real-world imbalanced datasets.

INDEX TERMS Incremental learning, network intrusion detection, softmax regressor, adaptive random forest, KNN classifier, majority voting classifier, random sampling, adaptive windowing.

I. INTRODUCTION

Intrusion detection plays a pivotal role in the current cybersecurity landscape as the number of evolving attacks, such as Distributed Denial of Service (DDoS), ransomware, and advanced persistent threats (APTs), [1], [2], [3] continues to grow on a daily basis. Security systems are in dire need of robust components that can effectively prevent potential attacks [4] and safeguard network integrity. IDS have emerged as indispensable tools for protecting systems against unauthorized access and mitigating the risks associated with malicious activities [5].

The associate editor coordinating the review of this manuscript and approving it for publication was Li Zhang .

In order to tackle above mentioned threats, The field of intrusion detection has witnessed numerous studies [6], [7], [8] employing both machine learning and deep learning approaches [9] to address the challenges in network security. While these studies have demonstrated favorable results in detecting diverse attack types, they are faced with two prominent issues. Firstly, the majority of existing IDS systems remain static, lacking the ability to adapt and learn in real-time. This limits their effectiveness in identifying newly emerging attacks and necessitates costly retraining processes. Secondly, these supervised models often rely on large, accurately labeled datasets, which can be laborious and expensive to obtain. Moreover, the storage requirements for such datasets can be substantial. These challenges highlight

the need for dynamic and resource-efficient intrusion detection approaches that can effectively handle evolving threats without extensive labeling efforts or storage constraints.

Therefore, Incremental learning approaches [10], [11] have gained significant attention in the field of intrusion detection due to their ability to learn and adapt continuously without the need for a complete and ready-to-go dataset [12]. These approaches allow models to be trained and launched with limited initial data and can continue to learn and update as new data becomes available. This flexibility makes them particularly suitable for dynamic environments where the data distribution and attack patterns may change over time.

However, one of the key challenges faced by incremental learning approaches is concept drift. Concept drift refers to the phenomenon where the statistical properties of the data, such as the underlying concepts or patterns, change over time [13]. This can occur due to various factors, including the emergence of new types of attacks or changes in the network environment. Concept drift poses a significant threat [14] to the accuracy and effectiveness of intrusion detection models, as the model's learned knowledge may become outdated and unable to accurately classify new instances.

Our research builds upon the work of CDIL [15], where a combined incremental model using majority voting [12], [16] was employed to tackle concept drift challenges in intrusion detection. However, in our experiment, we have achieved even better performance results. On top of that, Our approach effectively addresses evolving attacks and maintains high detection accuracy over time, demonstrating its potential as a valuable tool for real-world cybersecurity applications.

In order to train and evaluate the presented supervised algorithm, a suitable dataset is required. For this purpose, we utilize the CICIDS2017 [17] dataset provided by the Canadian Institute for Cybersecurity. This dataset is known for encompassing a wide range of prevalent attack types and is recognized as one of the recent and comprehensive datasets in the cybersecurity industry. Therefore, it served as a valuable choice for our research. However, it should be noted that the CICIDS2017 [17] dataset was mostly specialized for static models and had highly class imbalance. To address this issue, we performed preprocessing techniques to handle the imbalanced nature of the dataset and making it appropriate for sequential learning. These preprocessing steps were necessary to ensure fair and accurate training and evaluation of our presented algorithm.

A. CONTRIBUTION

This research paper introduces a significant contribution to the field of Intrusion Detection Systems (IDS) in the form of an Incremental Learning Based Majority Voting Model [12], [16]. Our model surpasses the performance of existing identical approaches, achieving an impressive accuracy rate of 96.43%, outperforming CDIL's [15] 94.91%, WDIS's [18] 96.32% and AB-HT [19].

Our model stands out by dynamically adapting to newly emerging attacks without the need for extensive

retraining using old data. This adaptation to concept drift is achieved through sophisticated machine learning techniques. By leveraging multiple algorithms, our model substantially enhances overall IDS accuracy. Furthermore, we employ Adaptive Windowing (ADWIN) [20], a powerful concept drift detection method, ensuring that our system remains current and effective against evolving attack patterns [13].

A notable feature of our approach is its cost-effectiveness in training with new datasets, thereby minimizing resource requirements and providing a scalable solution. In summary, this research offers a compelling and superior solution for practical intrusion detection compared to existing methods, addressing the evolving landscape of cybersecurity threats.

We begin by reviewing related works in Section II to establish the context. In Section III, we present the preliminaries, outlining the specific concepts which the reader has to get familiar with. Our presented approach is detailed in Section IV, followed by the experimental results in Section V. Finally, in the Conclusion, we summarize our findings and suggest future research directions.

II. RELATED WORK

The use of majority voting [16] has been widely explored by researchers in the field of intrusion detection. Numerous studies have been conducted to investigate the effectiveness of majority voting [21], [22], [23] as a classification technique for detecting intrusions.

In a notable contribution to the field, Mahendra and Aritsugi [19] devised an ensemble incremental learning model that amalgamated the strengths of AdaptiveBoosting (AdaBoost) and Hoeffding Tree (HT). Their primary objective was to alleviate the computational demands inherent to incremental learning models while maintaining robust performance standards. Their comprehensive experiments demonstrated that the ensemble model consistently outperformed the conventional HT and Hoeffding Any Time Tree (HATT) models. Most notably, the ensemble model achieved a significantly higher average F1-score. This achievement underscores the potential of their approach to streamline intrusion detection systems while enhancing their efficacy.

Bamhdi et al. [24] adopted an intrusion detection approach using a majority voting-based ensemble method to improve the detection rate by combining the results from multiple classifiers. To minimize training time and computational complexity, the researchers employed a decision tree algorithm to identify significant attributes from the dataset. By focusing on these important attributes, the model could achieve efficient processing while maintaining accuracy. The data was classified as either normal or intrusive using a one-is-to-many approach, which involved training the model to distinguish between different types of intrusions. To optimize the performance of the model, Particle Swarm Optimization (PSO) was employed.

Patil and Pattewar [25] applied an ensemble learning approach for intrusion detection, incorporating three different feature selection techniques: Gini Ratio, Information Gain,

and Correlation-based feature selection. They also utilized five machine learning algorithms, namely XGBoost, J48 Decision Tree, AdaBoost, Random Forest, and REPTree, for classification within the ensemble. They found that their majority voting ensemble outperformed the individual classification modules in terms of detection accuracy and performance metrics.

Yuan et al. [15] adopted an innovative method for intrusion detection by leveraging the concept drift detection capabilities of ensemble incremental learning. The authors address the challenges posed by dynamic and evolving intrusion patterns in real-world scenarios. Their approach combines ensemble learning with incremental learning techniques, allowing the model to adapt and update itself over time. By detecting and responding to concept drift, the proposed method ensures the accuracy and effectiveness of intrusion detection in dynamic environments. The paper contributes to the field by providing a comprehensive framework that tackles the evolving nature of intrusion patterns and offers a reliable approach to safeguarding network security.

Alotaibi and Elleithy [18] applied a framework for intrusion detection that utilized several machine-learning algorithms to build patterns for both normal behavior and intrusions. The framework consisted of an offline stage, where intrusion patterns were constructed, and an online stage, where intrusions were classified based on their types. The training stage included algorithms such as Extra Trees, Random Forests, and Bagging with Decision Trees, which were combined using majority voting to enhance robustness and achieve better results. Once the majority voting was applied, the patterns were constructed using a matching builder for both normal samples and intrusions. These patterns could be serialized and utilized in the detection capability of the system. In the online stage, network traces were pre-processed using the selected features identified during the feature selection stage. The pre-processed frames were then fed into the detection utility for real-time detection. The detection utility determined whether a frame was suspicious or not. If a suspicious frame was detected, an alert was triggered to notify the appropriate personnel of a potential intrusion.

Hao and Wang [26] introduced the concept of continuous few-shot learning for intrusion detection in their paper. They propose a metric-based first-order meta-learning framework, facilitating the training of intrusion detection models through multiple tasks. The framework aimed to maximize the model's generalization ability, enabling it to effectively handle various tasks. In the face of an expanding array of attack classes, their trained model exhibits the capability to promptly adapt to new attacks using only a few shots of samples. To mitigate knowledge loss, a repository of representative old-class attack samples is maintained. Extensive experiments demonstrate the model's commendable plasticity, showcasing its effectiveness in detecting new attacks with a minimal number of samples.

Tingting et al. [27] introduce ID-FSCIL, a learning strategy designed to address the few-shot class-incremental intrusion detection challenge. ID-FSCIL effectively extends the existing detection system to accommodate the growing number of attack categories. Leveraging meta-learning, it extracts new intrusion patterns from a minimal set of samples and optimizes the model's generalization ability to handle emerging attacks. Evaluations on the NSL-KDD dataset, conducted under incremental learning settings, demonstrate the superior performance of ID-FSCIL compared to state-of-the-art baselines.

Zheng et al. [28] proposed a two-level network intrusion detection method centered around ensemble learning. The first level employs a binary-classification model for swift determination of whether a network access behavior constitutes an attack. Meanwhile, the second level deploys a multi-classification model capable of categorizing anomalous access behaviors into specific attack types.

Constantinides et al. [29] introduced an innovative Network Intrusion Prevention System that leverages a Self-Organizing Incremental Neural Network in conjunction with a Support Vector Machine. The proposed system, owing to its unique structure, presents a security solution devoid of dependence on signatures or rules, exhibiting the capability to counter both known and unknown attacks in real-time with reasonable accuracy. Through experimentation with the NSL KDD dataset, their proposed framework demonstrates efficient and scalable industrial applications with its ability to achieve online updated incremental learning.

Ndichu et al. [30] paper proposes an online learning scheme tailored for critical threat alert detection to surmount these challenges. Treating threat alert data as a stream of items fed to the learning model enables a swifter and more responsive reaction to emerging threat alerts. Additionally, employing a focal loss function in learning effectively addresses the skewness prevalent in threat alert analysis scenarios. The proposed scheme undergoes evaluation on a benchmark dataset obtained from the security operation center of a large-scale enterprise network to identify potentially critical threat alerts.

Shahbandayeva et al. [8] proposed a hybrid approach which introduces a novel method for effective attack detection by combining supervised learning algorithms to identify known attacks and unsupervised learning for detecting unknown and zero-day attacks. They applied this approach to the CSE-CIC-IDS 2018 dataset, training classifiers to discern benign traffic and 14 known attacks using a set of 23 features. Instances where network traffic flows lacked a specific level of certainty in classification were directed to the clustering phase for identification as either benign or malicious traffic. Results indicate the success of the three classification algorithms—K-Nearest Neighbors, Random Forest, and Artificial Neural Networks—in classifying known attacks. Additionally, the clustering algorithm HDBSCAN demonstrated proficiency in successfully clustering

unclassified benign and malicious traffic with unknown labels.

Building upon the achievements of previous research in majority voting algorithms, our work introduces an enhanced and distinct approach. While acknowledging the contributions of these earlier studies, it is important to highlight the exceptional aspects of our model. As the demand for incremental models continues to rise, we have observed that most existing approaches are static [8], [19], [24], [25] and the incremental ones [15], [18], [28], [29] often on an excessive combination of algorithms, resulting in significant computational overhead or leverage single algorithms [26], [27], [30] which may result to false positives. In contrast, our model, which utilizes a concise ensemble of three carefully selected algorithms, stands out by offering incremental learning capabilities without compromising computational efficiency and detecting a greater number of classes compared to the aforementioned proposals. Additionally, our model incorporates a robust concept drift detection mechanism, enabling it to adapt to evolving data patterns and maintain high detection performance in dynamic environments. By proactively identifying and responding to concept drift, our model ensures the accuracy and reliability of intrusion detection over time. While some proposals [24], [25] may have slightly better accuracy, our model's ability to detect more classes and its optimized approach make it a compelling choice for practical intrusion detection.

III. PRELIMINARIES

In this section, we provide an overview of the key preliminary concepts and algorithms that form the foundation of our research in IDS based on Machine Learning. These fundamental approaches play a pivotal role in handling concept drift, optimizing model performance, and enabling adaptability to changing data scenarios. Before delving into the details of our presented Incremental Majority Voting approach, let us first explore the significance of these techniques in the context of IDS research.

A. Hoeffding Adaptive Tree(HAT)

The HAT algorithm, introduced by Bifet and Gavaldà in [31], is a decision tree-based method designed specifically for handling concept drift in streaming data analysis. It utilizes the Hoeffding bound, a statistical bound that allows for accurate and efficient learning from a small, randomly selected subset of the data.

HAT employs an incremental learning approach, where the decision tree is built and updated dynamically as new data arrives. Unlike traditional decision trees that require retraining from scratch, HAT incrementally adapts the existing tree structure by updating the statistics and probabilities at each tree node.

The Hoeffding bound plays a crucial role in HAT by determining the minimum number of instances needed to make an early decision. This enables HAT to make accurate

predictions with high confidence while minimizing the computational cost associated with tree updates.

One of the key advantages of HAT is its ability to handle concept drift in a timely manner. When a change in the data distribution is detected, HAT can adapt the decision tree by creating new branches or updating existing ones. This ensures that the model remains up-to-date and maintains its accuracy in evolving data scenarios.

B. ADAPTIVE RANDOM FOREST(ARF)

The ARF classifier, introduced by Gomes et al. [32], is an extension of the Random Forests ensemble method. Similar to Random Forests, the ARFs also utilize collections of weakly-correlated decision trees. However, it incorporates adaptive mechanisms to enhance its performance.

In ARFs, each decision tree in the ensemble is trained using a bootstrap sample of the training set. The key difference lies in the adaptive nature of the feature selection process. Rather than using a fixed random subset of features for each tree, ARFs dynamically adjust the subset of features considered for the best split at each node.

By adaptively selecting features, the ARFs algorithm ensures that each tree in the ensemble uses independent and diverse sets of features from the training samples. This adaptive feature selection procedure reduces the potential statistical correlations among the trees, thereby enhancing the ensemble's overall performance and robustness.

C. KNN CLASSIFIER

The KNN classifier [33] is a popular algorithm in machine learning for classification tasks. It is a non-parametric and instance-based learning method that makes predictions based on the similarity between input samples.

In our research, we specifically configured the KNN algorithm with the number of neighbors set to 8. During prediction, the KNN classifier identifies the 8 nearest neighbors to the input sample using a chosen distance metric, such as Euclidean distance. The class label of the input sample is determined through majority voting among these 8 neighbors.

One of the significant advantages of the KNN classifier is its simplicity and ease of implementation. It does not require an explicit training process, as it stores the entire training dataset for reference during prediction. Moreover, the KNN algorithm is non-parametric, meaning it does not make assumptions about the underlying data distribution.

D. SOFTMAX REGRESSOR

The Softmax Regressor, also known as Multinomial Logistic Regression, is a popular algorithm in machine learning for classification tasks. It is particularly suitable for problems with multiple classes, where the goal is to assign an input to one of the available classes.

The Softmax Regressor extends the logistic regression model to handle multiple classes by using the Softmax function. This function takes the outputs of the linear

regression [34] model and transforms them into a probability distribution over the classes. Each class is assigned a probability value, indicating the likelihood of the input belonging to that class.

During training, the Softmax Regressor learns the optimal weights and biases that minimize the cross-entropy loss between the predicted probabilities and the true class labels. It employs iterative optimization techniques, such as gradient descent, to update the model parameters and improve its predictive performance.

One of the key advantages of the Softmax Regressor is its interpretability. The learned weights can be examined to understand the influence of different features on the classification decision. Additionally, the predicted probabilities provide insights into the model's confidence in its predictions.

The Softmax Regressor has applications in various domains, including natural language processing, image classification, and multi-class classification problems in general. Its ability to handle multiple classes and provide interpretable results makes it a valuable tool in the machine learning toolkit.

E. MAJORITY VOTING

The Majority Voting classifier is a popular ensemble learning method that combines predictions from multiple base classifiers to make final decisions. It is based on the principle that aggregating the predictions of several models can lead to more accurate and robust predictions compared to using a single model.

In Majority Voting, each base classifier in the ensemble independently predicts the class label for a given input sample. The class label that receives the majority of votes from the base classifiers is chosen as the final prediction. In the case of a tie, various tie-breaking strategies can be employed, such as selecting the class label with the highest confidence score or choosing randomly.

The strength of the Majority Voting classifier lies in its ability to leverage the diversity among the base classifiers. When the base classifiers have different strengths and weaknesses or are trained on different subsets of the data, the ensemble can capture a broader range of patterns and make more accurate predictions. This diversity helps to reduce bias and variance in the predictions, leading to improved overall performance.

The Majority Voting classifier is particularly effective when the base classifiers are diverse and independent, meaning they make errors on different samples or in different regions of the feature space. By combining their predictions, the ensemble can achieve higher accuracy and better generalization.

One advantage of Majority Voting is its simplicity and ease of implementation. It can be applied to various classification algorithms, such as decision trees, logistic regression, or support vector machines, making it a versatile ensemble method.

F. ADAPTIVE WINDOWING (ADWIN)

Adaptive Windowing, introduced by Bifet et al. [20], is a method for handling concept drift in streaming data analysis. It extends the traditional sliding window approach by dynamically adjusting the window size based on the detected changes in the data distribution.

In Adaptive Windowing, the window size is initially set to a fixed value. As new data arrives, the algorithm continuously monitors the data distribution within the window. When a significant change in the distribution is detected, indicating a potential concept drift, the window size is adjusted to accommodate the new data distribution. The adjustment of the window size is performed based on predefined criteria, such as the magnitude of the distribution change or the accuracy of the existing model. A larger window size may be used to capture a more stable data distribution, while a smaller window size can adapt to rapid changes in the data. By adapting the window size, Adaptive Windowing enables the model to effectively adapt to concept drift and maintain its performance over time. It allows for better tracking of evolving data patterns and improves the model's ability to capture new concepts or adapt to changing conditions.

The adaptive nature of the window size selection in Adaptive Windowing enhances the model's robustness and adaptability in dynamic environments. It provides a flexible framework for continuously monitoring and updating the model to handle concept drift effectively.

G. NO FREE LUNCH THEOREM

The No Free Lunch (NFL) theorem, introduced by Wolpert and Macready in 1997 [35], is a fundamental concept in machine learning and optimization. It states that, on average, all optimization algorithms perform equally across all possible problem domains. This challenges the idea of a universally superior algorithm and emphasizes the importance of algorithm selection and adaptation based on the specific problem at hand. The NFL theorem highlights the need to consider the problem's characteristics, such as data distribution, problem structure, and optimization criteria, when choosing an algorithm. It reminds us that different algorithms may excel in different domains, and there is no one-size-fits-all solution. This calls for continuous research and development of tailored algorithms to address the unique challenges of specific problem domains, leading to more effective and efficient solutions.

IV. PROPOSED METHOD

The main objective of this research is to develop an effective intrusion detection system with a low false positive rate using ensemble learning techniques. In this study, we present a classifier composed of three machine learning classifiers selected from different families. The choice of classifiers is motivated by the "no free lunch" theorem, aiming to leverage models that complement each other during the classification process. Initially, two models illustrated in Table 1 were

TABLE 1. Constructed models with classifiers.

Model	Classifier 1	Classifier 2	Classifier 3
Model 1	ARF	Softmax Regressor	KNN
Model 2	HAT	Softmax Regressor	KNN

constructed using combinations of ARF, Softmax Regressor, and KNN, as well as HAT, Softmax Regressor, and KNN. The performance of these models was evaluated, and the one demonstrating exceptional performance was selected for further analysis. Our model incorporates concept drift detection and resolution as an integral feature, leveraging the ADWIN algorithm implemented in the river [36] library. This default implementation empowers our model to automatically detect and adapt to changes in data distributions without requiring manual configuration or adjustments.

In our presented model, the chosen classifiers work simultaneously, and their individual predictions are combined using the Weighted Majority Voting method to obtain the final classification result. The complete process is illustrated in Figure 1, providing a visual representation of the ensemble learning approach utilized in this research.

A. DATASET

In our research, the CICIDS2017 [17] dataset was chosen for its reliability and comprehensive coverage of multiple types of attacks. This dataset has proven to be a valuable resource for researchers as it allows them to evaluate their models using the latest attack types.

The CICIDS2017 [17] dataset consists of 84 network features that were extracted using the CICFlowMeter software provided by the Canadian Institute for Cybersecurity. To ensure the dataset’s alignment with real-world scenarios, we followed a similar approach as previous studies [37] and removed six features: Flow ID, Protocol, Timestamp, Source IP, Destination IP, and Source Port. These features may have values that differ from real-world scenarios because the dataset was generated in an isolated network environment. By utilizing the CICIDS2017 [17] dataset, we were able to conduct our research using a reliable and up-to-date dataset that captures various types of network intrusions.

Before training our ensemble model, we performed several preprocessing stages on the dataset to ensure its quality and suitability for our research. These preprocessing steps included scaling the features to normalize their values and removing the rows with missing column in the data. Given that the dataset was not specifically designed for incremental learning, we divided it into batches. Each batch contained different categories of traffic from each other, enabling us to structure the data into manageable subsets. This batch splitting approach allowed us to effectively train our ensemble model on the dataset while accommodating its non-incremental nature. Table 2 represents the final dataset obtained after preprocessing. After preprocessing the dataset, we further selected a subset, represented in Figure 2

TABLE 2. Dataset after preprocessing.

Batch	New Label	Initial Labels	Number of Instances
1	Benign	Benign	529918
2	Dos	Dos-Goldeneye, Dos-Hulk, Dos-Slowhttptest, Heartbleed	252672
3	DDos	DDos	128027
4	Brute Force	FTP-Patator, SSH-Patator	13835
5	PortScan	PortScan	158930
6	Web Attack	Web Attack-Brute Force, Web Attack-Sql Injection, Web Attack-XSS	2180
7	Bot	Bot	1966

that maintained the same label distribution. This subset was specifically chosen for conducting our experiments. To evaluate the practical performance of our model, we divided the dataset into training and testing parts using an 80/20 ratio after shuffling.

B. FEATURE EXTRACTION

Feature selection is a critical step in machine learning and data analysis, aimed at identifying the most relevant and informative features from a given dataset. By selecting a subset of features that are highly correlated with the target variable, feature selection helps to reduce the dimensionality of the data and improve the performance of predictive models. It not only enhances the accuracy of the models but also reduces the computational complexity and training time. Various techniques for feature selection exist, ranging from statistical methods such as correlation analysis and chi-square tests to more advanced approaches like recursive feature elimination and genetic algorithms. The selection of appropriate features is essential for building robust and interpretable models, as it focuses on the most meaningful aspects of the data and reduces the impact of noise and irrelevant information.

To streamline the computation and enhance the efficiency of our model, we employed the feature importance method of the Random Forest regressor. This allowed us to identify and select the 25 most relevant features out of 79 from the dataset. The selected features, as presented in Table 3, were found to significantly contribute to the model’s predictive performance. By focusing on these key features, we observed a remarkable improvement in computational speed, achieving a threefold increase compared to using the original feature set. This feature selection process enabled us to streamline our analysis and expedite the training and evaluation of our model.

C. HYPERPARAMETER SELECTION

While using machine learning models, choosing optimal hyper-parameters is a crucial task. Thus, we conducted

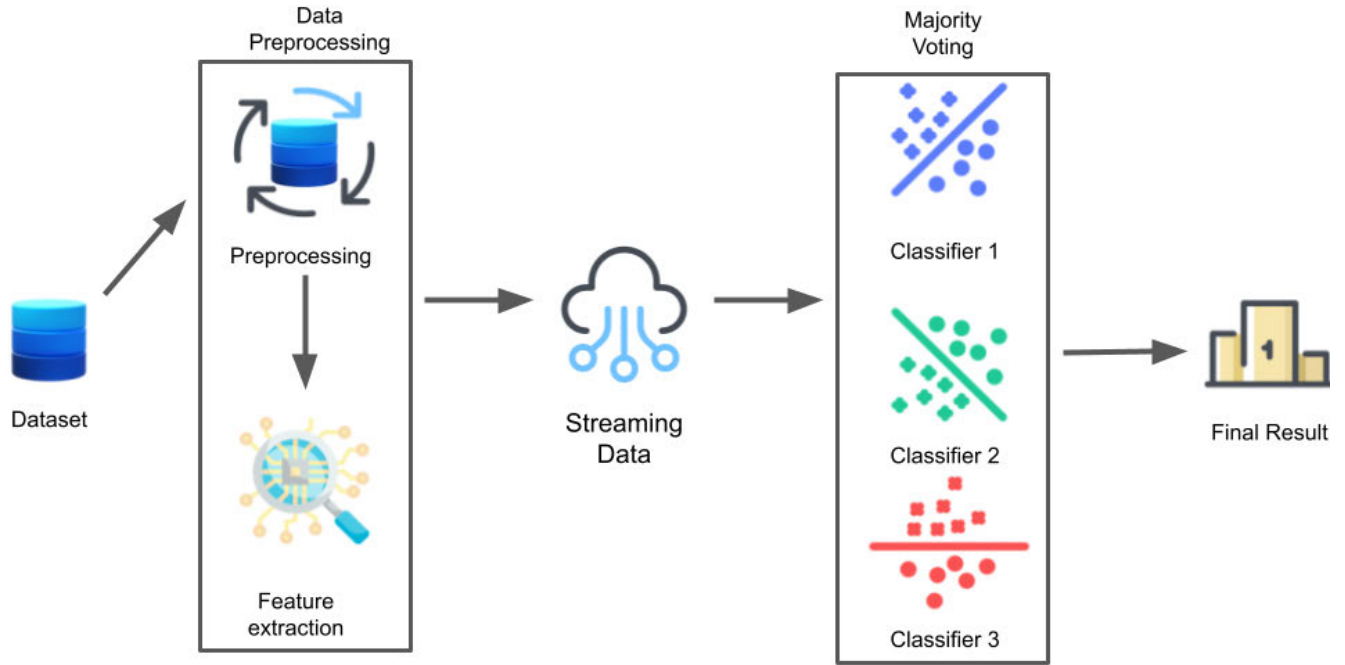


FIGURE 1. Proposed model scheme.

TABLE 3. Top 25 features.

No	Feature name
1	min packet length
2	init win bytes backward
3	destination port
4	bwd packets/s
5	average packet size
6	psh flag count
7	init win bytes forward
8	flow iat min
9	fwd iat min
10	packet length std
11	act data pkt fwd
12	fwd iat max
13	bwd iat min
14	bwd packet length std
15	fwd iat total
16	fwd packets/s
17	flow duration
18	total length of fwd packets
19	idle std
20	max packet length
21	subflow fwd bytes
22	total length of bwd packets
23	fwd packet length mean
24	active max
25	fwd packet length std

an in-depth investigation into the utilization of HAT with customized settings. To optimize the performance of HAT, we carefully configured the leaf prediction attribute to Naive Bayes Adaptive and selected Info Gain(Eq. 1) as the split criterion. Furthermore, we set the delta to 1e-5 and tau to 0.099, fine-tuning these parameters for optimal performance. Through rigorous experimentation, we discovered that these

specific configurations yielded exceptional performance and delivered outstanding outcomes for our research scenario. By leveraging Naive Bayes Adaptive for leaf prediction, utilizing the Info Gain(Eq. 1) split criterion, and optimizing the delta and tau values, we aimed to enhance the accuracy and effectiveness of the HAT classifier, resulting in improved handling of our dataset.

$$IG(D, A) = H(D) - H(D|A) \tag{1}$$

where:

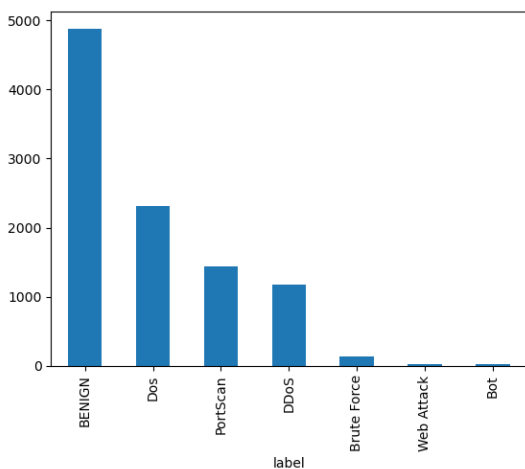
- D represents the dataset.
- A is an attribute or feature used for splitting.
- $H(D)$ is the entropy of the dataset D .
- $H(D|A)$ is the conditional entropy of D given attribute A .

In addition to customizing the HAT classifier, we further enhanced the performance of ARF by selecting specific hyperparameter settings. By setting the leaf prediction attribute to Naive Bayes Adaptive and utilizing the Hellinger(Eq. 2) split criterion, we optimized the ARFs classifier for our specific case. Furthermore, we set the tau value to 0.99 and the delta value to 1e-5, fine-tuning these parameters to achieve optimal performance. These configurations significantly improved the accuracy and effectiveness of the ARFs classifier, enabling it to handle our dataset more effectively.

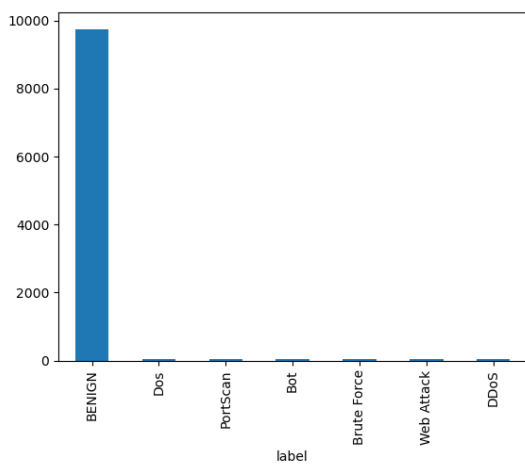
$$HD(P, Q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^n (\sqrt{p_i} - \sqrt{q_i})^2} \tag{2}$$

where:

- P and Q are probability distributions.



(a) Conventional Label Distribution



(b) Realistic Label Distribution

FIGURE 2. Comparison of class distributions.

- p_i and q_i represent the probabilities of individual events or outcomes in the distributions.

The performance of the KNN classifier heavily relies on the choice of the parameter K, representing the number of neighbors considered for prediction. After careful experimentation, we determined that setting K to 8 resulted in optimal performance. This choice allowed the KNN classifier to establish flexible decision boundaries while remaining robust to noise and effectively capturing local patterns in the data.

V. EXPERIMENTAL SCENARIOS

In the realm of incremental learning, our model adapts and learns continuously as data flows in. During our experiments, we crafted two distinct scenarios to meticulously assess the reliability of our model. In the first scenario, we employed a conventional approach to partition the dataset. This involved ensuring that all label distributions were meticulously balanced, with ample samples for each class to robustly train the model. Moving to the second scenario, our focus shifted to

simulating a real-world setting. Here, we deliberately skewed the dataset to mirror a common occurrence: the benign class dominating with a majority of samples, while the attack labels represented a minority.

A. CONVENTIONAL LABEL DISTRIBUTION

During the first scenario, we aimed to demonstrate the superiority of our constructed models compared to other approaches in handling incremental learning. To achieve this, we conducted experiments in various scenarios to assess the performance of our constructed models by evaluating their performance under different conditions.

The research process involved several steps. Initially, we set specific hyper-parameters for the algorithms and trained both ensemble models. We evaluated the performance of the models using relevant metrics to assess their capability in handling incremental learning.

After training our ensemble models, we proceeded to evaluate the performance of each individual algorithm within the ensembles. We split the ensemble models into their constituent algorithms and trained each algorithm separately using the same dataset. This allowed us to directly compare the performance metrics of the individual classifiers with those achieved by the ensemble models, while ensuring a fair and consistent experimental setup.

By comparing the metrics obtained from training the individual classifiers with the metrics obtained from training the ensemble models, we aimed to demonstrate the superiority of the ensemble approach over the individual classifiers. This comparison enabled us to highlight the enhanced performance and effectiveness of the ensemble models when trained with the same hyper-parameters

In addition to our previous experiments, we conducted another experiment to assess whether the current hyper-parameters used for the individual classifiers were optimal or could be further improved through fine-tuning. The results of this experiment revealed that fine-tuning the hyper-parameters of the single classifiers did lead to some performance improvements. However, it is important to note that these improvements were not substantial enough to surpass the performance achieved by our ensemble model.

In the context of working with imbalanced and streaming data, it is crucial to address the inherent imbalance to ensure accurate and reliable results. One widely adopted approach is random sampling, which has proven to be highly effective in improving performance in such scenarios. Building upon this premise, we integrated random sampling, along with under-sampling and over-sampling techniques, into our dataset preprocessing pipeline.

By incorporating these sampling techniques, we aimed to assess their suitability and potential for enhancing the performance of our model in handling imbalanced streaming data. This comprehensive evaluation allowed us to gain valuable insights into the effectiveness of random sampling and its impact on our specific research context.

Lastly, To demonstrate the exceptional qualities and distinctive contributions of our model, we have planned a comprehensive comparative analysis with recently established models. This analysis will be conducted as the final step in our experimental scenarios. We have thoughtfully chosen two well-established models [15], [18] that closely align with our research objectives, along with one model [19] that exhibits remarkable similarity to our specific case with its incremental nature and dataset used. All three of these models have garnered substantial attention and recognition within the field.

B. REALISTIC LABEL DISTRIBUTION

In the implementation of the real-world scenario, we meticulously partitioned our dataset as depicted in Figure 2. This partitioning reflects the inherent characteristics of a real-world setting, where benign traffic predominantly outweighs instances of attacks, which occur relatively infrequently. The decision to create this scenario was rooted in the understanding that, in real-world scenarios, benign traffic tends to be the norm, with attacks occurring sporadically. Given that our model operates in real-time, learning dynamically from incoming traffic, simulating such a scenario provides a robust test environment to evaluate its efficacy under conditions that mirror actual traffic patterns. Subsequently, we subjected our primary proposed model to rigorous testing using diverse datasets, further substantiating its reliability across different contexts.

C. METRICS

In our study, we employed accuracy (Eq. 3), precision (Eq. 4), recall (Eq. 5), and F1-score (Eq. 6) as the primary classification metrics, as well as, false alarm metric was also calculated for the final outperforming model. To ensure a comprehensive evaluation, we considered the “pred” attribute of the confusion matrix, which provided insights into the models’ predictive capabilities and their alignment with ground truth labels. This allowed for a robust assessment of their performance in classification tasks.

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Population}} \quad (3)$$

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad (4)$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad (5)$$

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (6)$$

$$\text{FAR} = \frac{\text{False Alarms}}{\text{Total Alarms}} \quad (7)$$

VI. EXPERIMENTAL RESULTS

A. CONVENTIONAL DATASET PARTITION EXPERIMENTAL RESULTS

In this section, we provide a detailed analysis of our ensemble approach’s performance using machine learning models on

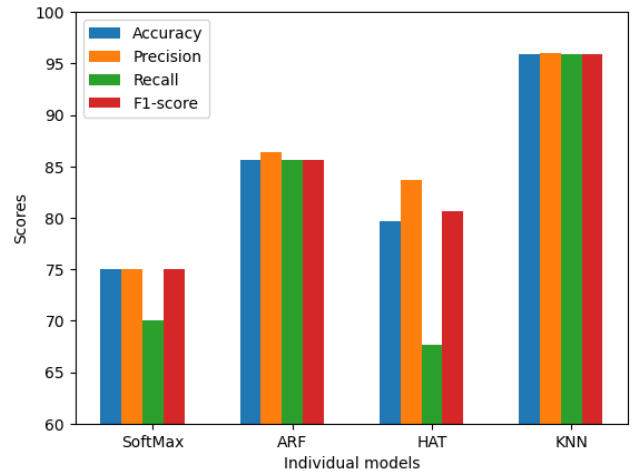


FIGURE 3. Results of fine-tuned individual classifiers.

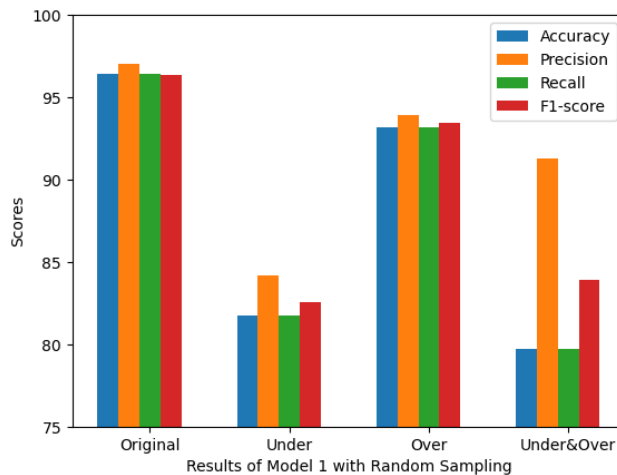
the CICIDS 2017 dataset. The results, as shown in Figure 8, offer a comprehensive view of the performance of both the individual base classifiers and the ensemble models. Our ensemble models consistently outperformed the single classifiers, achieving an impressive overall performance of 96.43% and 93.35% across all evaluation metrics, while the single classifiers lagged significantly with results of 58.72%, 83.98%, and 70.38%, respectively. It’s crucial to highlight that our KNN classifier showed competitive performance, achieving 95.88%, approaching the accuracy of our best model. For a more granular analysis, we evaluated the prediction accuracy for individual classes. Figure 17 (see Appendix for detailed figures) illustrates that our Model 1 consistently outperformed all other approaches in our experiment.

To further enhance our model’s performance, we engaged in fine-tuning efforts for the individual classifiers. These efforts resulted in notable performance improvements, ranging from 2% to 18%, as demonstrated in Figure 3. These improvements were achieved by fine-tuning hyperparameters specifically tailored to the single classification case. Importantly, even with these individually optimized classifiers, our ensemble model maintained its superior predictive accuracy, underscoring its robustness and effectiveness in capturing and leveraging the collective knowledge of the individual classifiers. This detailed analysis sheds light on the reasons behind our method’s strong performance and highlights the advantages of our ensemble approach in intrusion detection.

In our pursuit to address the challenge of imbalanced streaming data, we conducted a comprehensive set of experiments employing three pivotal random sampling techniques: Under Sampling, Over Sampling, and a Hybrid Sampling

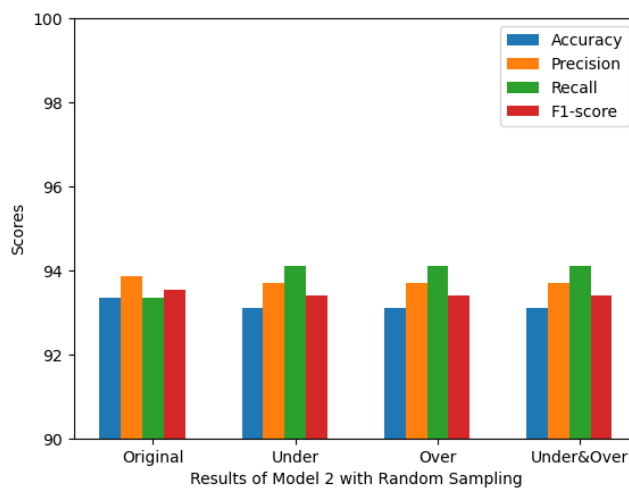
approach that combines elements of both. These methods are well-established in the field for their effectiveness in managing imbalanced datasets. Under ‘Under Sampling,’ we intentionally decreased the representation of majority classes to create a more balanced distribution, ensuring that minority classes receive adequate attention. This was accomplished by randomly selecting a representative subset of instances from the majority class while considering the desired distribution, such as reducing the majority class (0) to 20%, class 1 to 10%, class 2 to 15%, class 3 to 15%, class 4 to 10%, class 5 to 20%, and class 6 to 10% to make distribution more balanced. Conversely, ‘Over Sampling’ involved augmenting the instances in minority classes through various techniques like duplication or synthetic sample generation. This augmentation aimed to equalize class distributions and mitigate bias toward the majority class. The ‘Hybrid Sampling’ technique skillfully merged under and over-sampling, seeking a delicate equilibrium between bolstering minority class representation and mitigating the influence of majority classes. This approach was thoughtfully designed to harness the strengths of both techniques. Importantly, considering the incremental nature of our model and the continuous inflow of data, these sampling techniques are dynamically applied as every sample enters the model for learning. Furthermore, label distributions are adjusted in real-time based on a predefined rate. This meticulous design enhances the adaptability of our model to the ever-evolving data landscape. Remarkably, our best-performing model exhibited some performance fluctuations, resulting in a modest accuracy reduction to 79.7%, as illustrated in Figure 4 and Figure 5. In stark contrast, Model 2 displayed consistent and reliable performance, with minimal metric variations. This observation hints at Model 1’s heightened sensitivity to data fluctuations compared to Model 2, a trait that could prove advantageous in scenarios necessitating real-time concept drift detection and adaptability to changing data patterns. However, it is imperative to note that the application of random sampling techniques resulted in a noticeable decline in prediction accuracy for both models. We acknowledge that the effectiveness of these methods may fluctuate based on the dataset’s characteristics and the specific problem context. In our specific context, the implementation of random sampling, over-sampling, and under-sampling did not yield the anticipated performance enhancements.

Figure 6 showcases a comprehensive comparison between our best performing model and recent state-of-the-art papers [15], [18], [19]. Notably, our model demonstrates exceptional performance with the highest accuracy of 96.39% compared to the other incremental ensemble learning approaches, which achieve scores of 94.91% and 96.32% respectively. While the CDIL approach shows slightly higher Precision and Recall values, it is crucial to consider that our model excels in handling a greater number of class types. Although F1-score metrics were not available for the compared models [15], [18], the overall superiority of our model is evident coming from the obtained results.



Methods	Accuracy	Precision	Recall	F1-score
Original Performance	96.43%	97.01%	96.43%	96.38%
Random Under Sampler	81.76%	84.20%	81.76%	82.53%
Random Over Sampler	93.19%	93.89%	93.19%	93.44%
Random Under/Over Sampler	79.7%	91.31%	79.7%	83.89%

FIGURE 4. Result of Model 1 with random sampler.



Methods	Accuracy	Precision	Recall	F1-score
Original Performance	93.35%	93.85%	93.35%	93.55%
Random Under Sampler	93.10%	93.71%	94.10%	93.39%
Random Over Sampler	93.10%	93.71%	94.10%	93.39%
Random Under/Over Sampler	93.10%	93.71%	93.10%	93.39%

FIGURE 5. Result of Model 2 with random sampler.

Additionally, within the same table, we present the performance results of our recently developed AB-HT models. These models share a similar approach, involving incremental

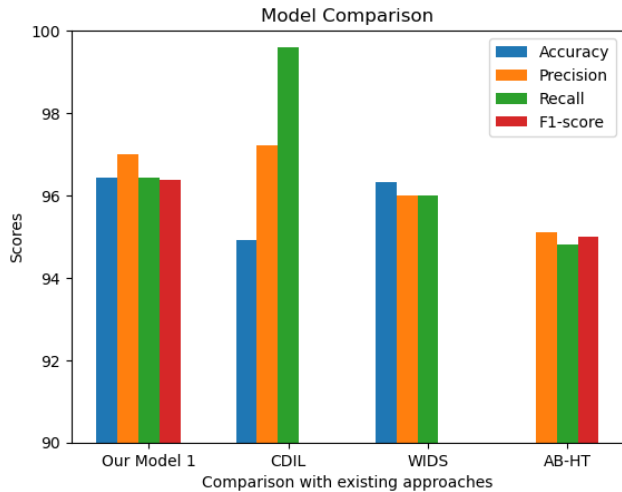


FIGURE 6. Comparison of our approach with already existing methods.

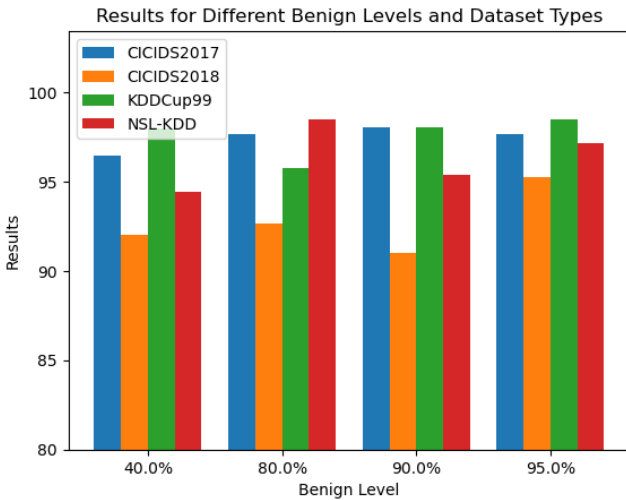


FIGURE 7. Results for different Benign levels and dataset types.

learning and ensemble techniques, using the CICIDS2017 [17] dataset. Notably, our model performed favorably across all the presented metrics in this comparison. While the precision and recall obtained by AB-HT are commendable at 95.1% and 94.8%, respectively, our model achieved slightly higher results with precision at 97.01% and recall at 96.43%. We are pleased with our model’s performance, which indicates its potential as a valuable solution in the field.

Furthermore, to thoroughly assess the robustness and generalization capabilities of our model, we conducted

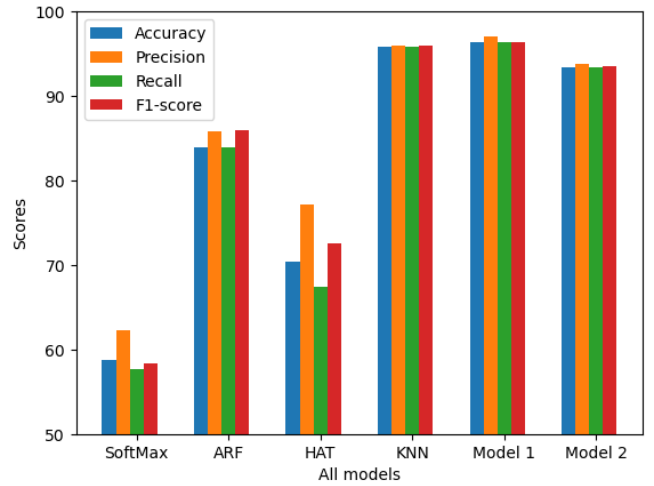


FIGURE 8. Results of individual and ensemble models.

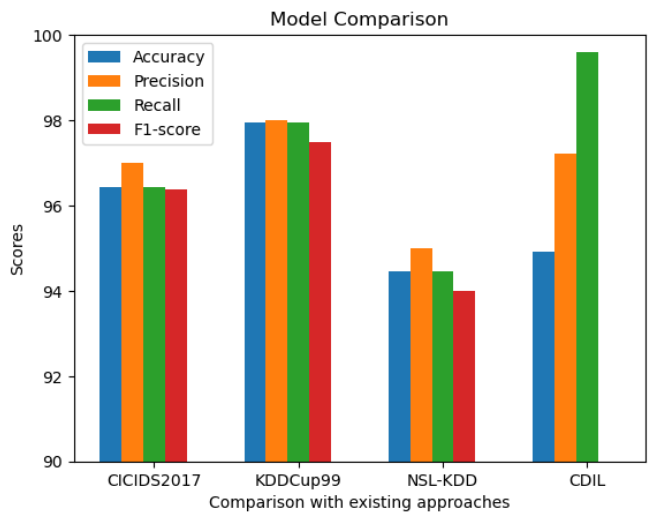


FIGURE 9. Our Model vs CDIL.

additional experiments on two distinct datasets: NSL-KDD and KDDCup99 [38]. Notably, KDDCup99 was prominently featured in CDIL’s [15] experiments, adding an extra layer of

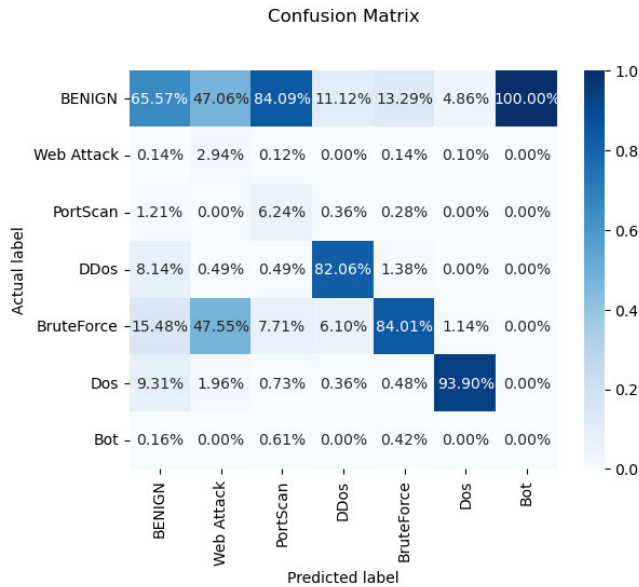


FIGURE 10. HAT.

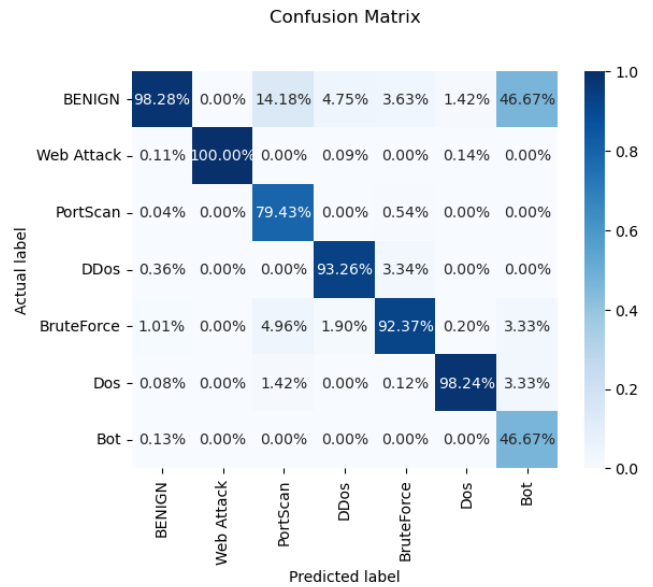


FIGURE 12. KNN classifier.

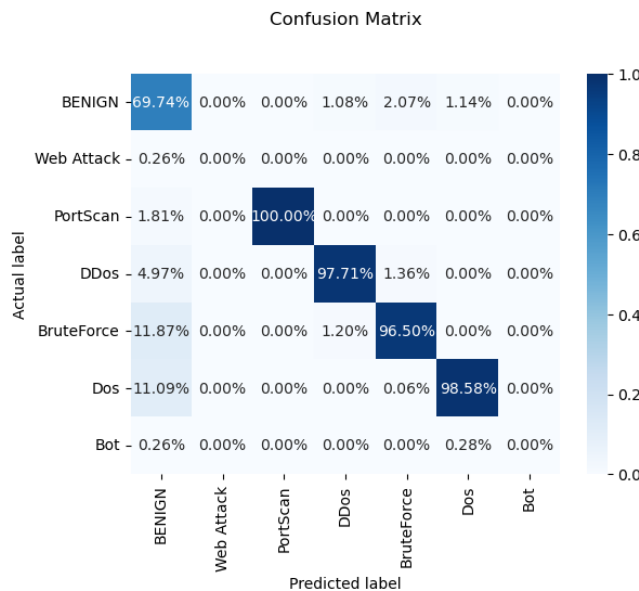


FIGURE 11. ARF.

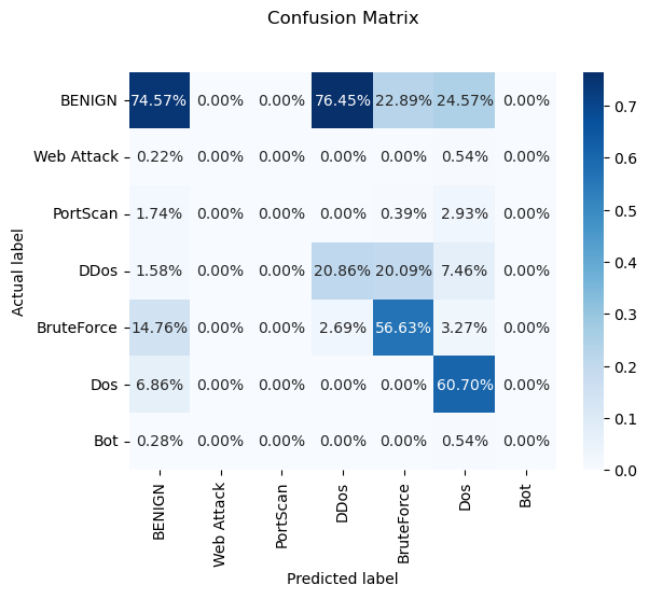


FIGURE 13. Softmax regressor.

comparison. Importantly, we maintained the integrity of our models and applied them directly to these datasets without any modifications. The outcomes were striking, as our model exhibited exceptional performance on the KDDCup99 dataset, achieving an impressive accuracy of 97.94%. This result surpassed the performance on our original datasets, reinforcing the versatility of our model.

Similarly, our model's performance on the NSL-KDD dataset was highly promising, yielding an accuracy of 94.45% and a precision rate of 95%. These results, obtained from different datasets, further validate our model's robustness and generalization capabilities.

B. REAL WORLD DATASET PARTITIONING RESULTS

In the concluding phase of our experiment, we sought to assess the adaptability of our proposed model without any modifications. Specifically, we endeavored to simulate a real-world scenario by reshaping the dataset, and testing in different rates of BENIGN classes starting from 40% until 95%. This realignment aimed to test the model's performance under conditions where benign instances dominate. The datasets utilized for this evaluation included CICIDS-2017, CICIDS-2018 [17], KDDCup2019, and NSL-KDD.

Throughout the experiment, the dataset was streamed to the model, allowing it to learn dynamically in real-time. The

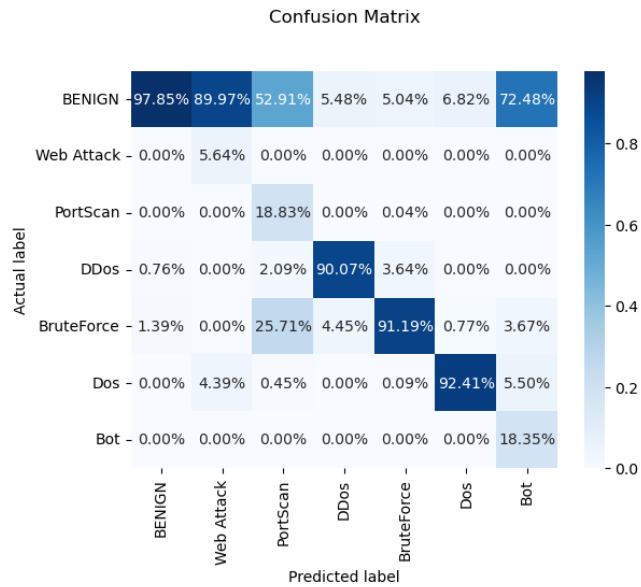


FIGURE 14. Model 1 random sampler.

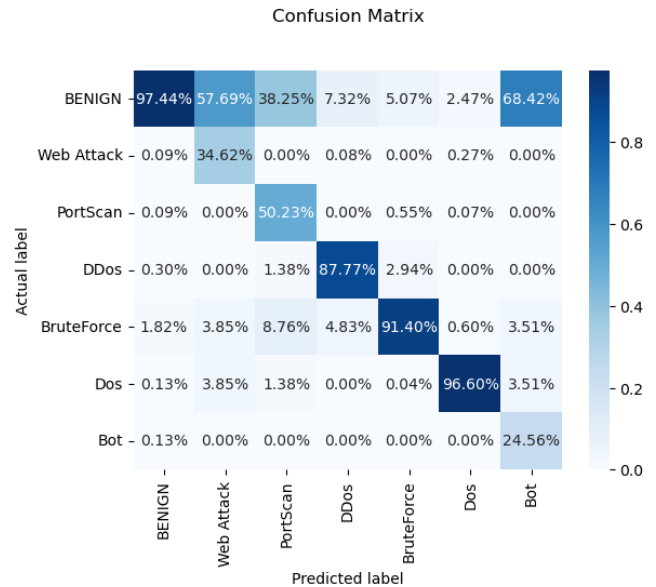


FIGURE 16. Model 1 over sampler.

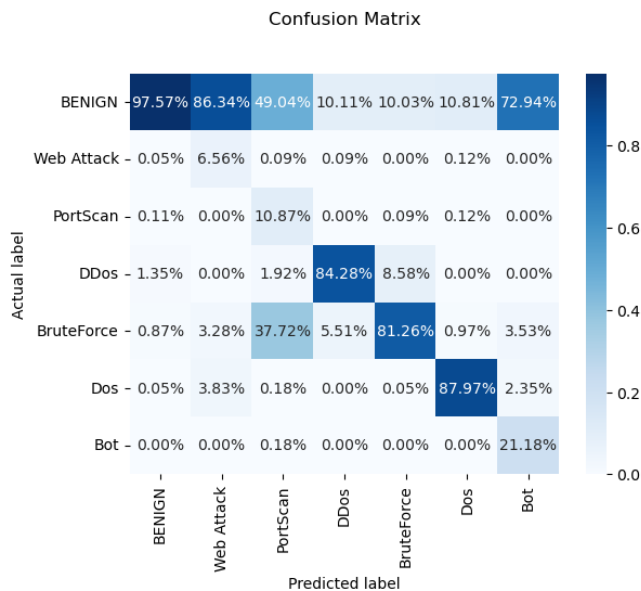


FIGURE 15. Model 1 under sampler.

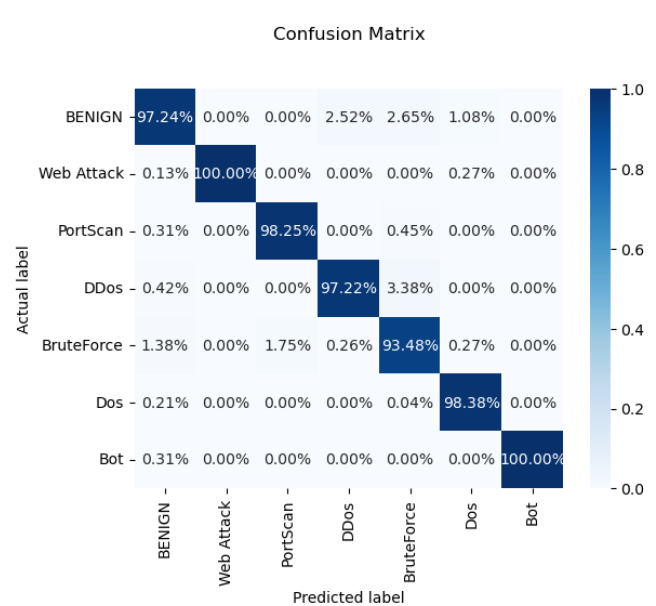


FIGURE 17. Model 1 best performance.

results, as summarized in Figure 7, demonstrated the model’s robust performance in various rates of BENIGN samples. Notably, it achieved a maximum accuracy of 98.48%, with commendable results of 97.68% and 95.78% for NSL-KDD, CICIDS-2017, and KDDCup99 respectively over the whole rate levels. Finally concluded with an accuracy of 95.24% for CICIDS-2018.

VII. CONCLUSION

In conclusion, our research addresses the limitations of traditional static and single classifier machine learning algorithms in detecting intrusion attacks by proposing an Incremental Ensemble Learning (IEL) approach. By focusing

on concept drift in continuous data streams and utilizing the CICIDS2017 dataset [17], we have achieved high accuracy and outstanding prediction accuracy in attack detection. Through our extensive experimentation with HAT, ARF, KNN, and Softmax Regressor algorithms, as well as the ensemble model constructed using these algorithms, we have demonstrated the superiority of the voting-based IDS model. This model surpasses individual classifiers and achieves a commendable detection rate.

Notably, our best-performing model achieved an overall accuracy of around 96.43%, with a remarkable prediction accuracy score of 100% for Web Attack and Bot attack

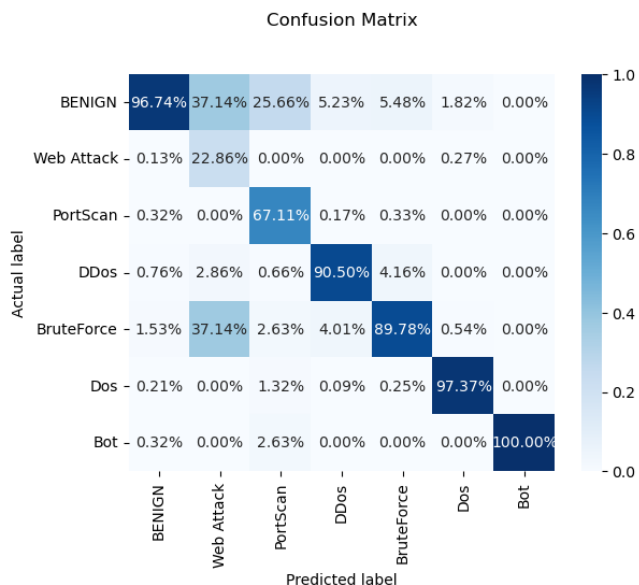


FIGURE 18. Model 2 random sampler.

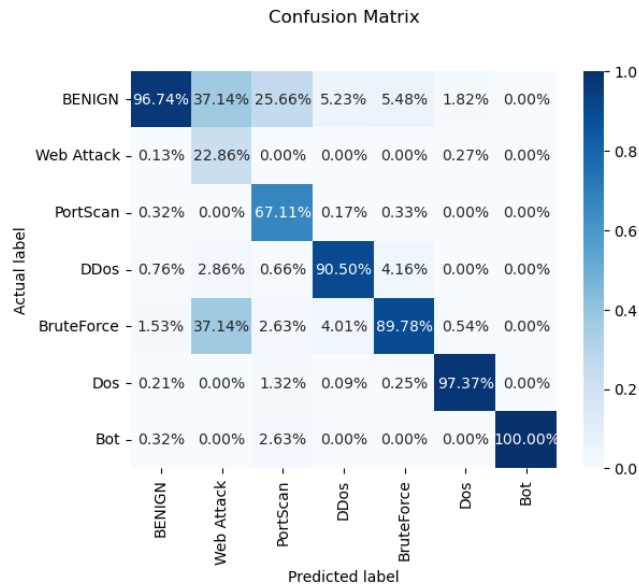


FIGURE 20. Model 2 over sampler.

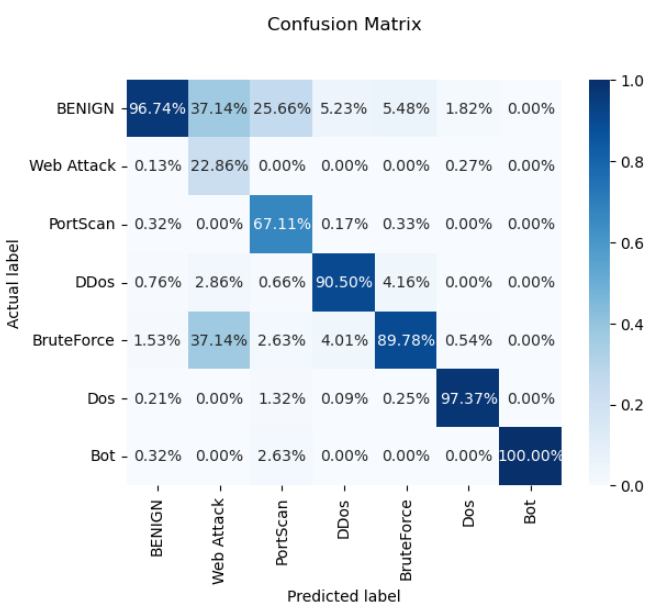


FIGURE 19. Model 2 under sampler.

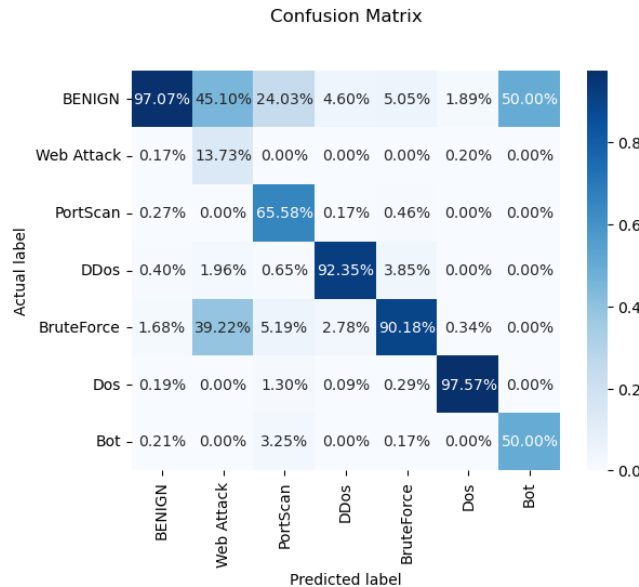


FIGURE 21. Model 2 best performance.

categories. Even though its performance was slightly lower for bot attacks, we have established a solid foundation for future enhancements and improvements. Importantly, our model achieved a remarkably low False Alarm Rate (FAR) of only 2%, underlining its ability to distinguish between legitimate and potentially malicious activities effectively.

In future research, our focus will be on further refining the model to increase its robustness in detecting new attacks. By continuing our efforts, we aim to push the boundaries of IDS and achieve even higher detection scores while maintaining a low FAR, contributing to the advancement of intrusion detection technology.

APPENDIX

See Figures 10–21.

REFERENCES

- [1] M. Mijwil, O. J. Unogwu, Y. Filali, I. Bala, and H. Al-Shahwani, "Exploring the top five evolving threats in cybersecurity: An in-depth overview," *Mesopotamian J. Cyber Secur.*, vol. 2023, pp. 57–63, Mar. 2023.
- [2] T. Fadziso, U. Thaduri, S. Dekkati, V. Ballamudi, and H. Desamsetti, "Evolution of the cyber security threat: An overview of the scale of cyber threat," *Digitalization Sustainability Rev.*, vol. 3, no. 1, pp. 1–12, 2023.
- [3] R. Dillon, P. Lothian, S. Grewal, D. Pereira, and A. Kuah, "Cyber security: Evolving threats in an ever changing world," in *Digital Transformation in a Post-Covid World: Sustainable Innovation, Disruption and Change*. Boca Raton, FL, USA: CRC Press, 2021, pp. 129–154.

- [4] P. Vanin, T. Newe, L. L. Dhirani, E. O'Connell, D. O'Shea, B. Lee, and M. Rao, "A study of network intrusion detection systems using artificial intelligence/machine learning," *Appl. Sci.*, vol. 12, no. 22, p. 11752, Nov. 2022, doi: [10.3390/app122211752](https://doi.org/10.3390/app122211752).
- [5] H. Albasheer, M. Md Siraj, A. Mubarakali, O. Elsier Tayfour, S. Salih, M. Hamdan, S. Khan, A. Zainal, and S. Kamarudeen, "Cyber-attack prediction based on network intrusion detection systems for alert correlation techniques: A survey," *Sensors*, vol. 22, no. 4, p. 1494, Feb. 2022.
- [6] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 1, p. e4150, Jan. 2021.
- [7] H. Liu and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: A survey," *Appl. Sci.*, vol. 9, no. 20, p. 4396, Oct. 2019, doi: [10.3390/app9204396](https://doi.org/10.3390/app9204396).
- [8] L. Shahbandayeva, U. Mammadzada, I. Manafova, S. Jafarli, and A. Z. Adamov, "Network intrusion detection using supervised and unsupervised machine learning," in *Proc. IEEE 16th Int. Conf. Appl. Inf. Commun. Technol. (AICT)*, Oct. 2022, pp. 1–7.
- [9] S. Gamage and J. Samarabandu, "Deep learning methods in network intrusion detection: A survey and an objective comparison," *J. Netw. Comput. Appl.*, vol. 169, Nov. 2020, Art. no. 102767. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1084804520302411>
- [10] K. S. Adewole, T. T. Salau-Ibrahim, A. L. Imoize, I. D. Oladipo, M. AbdulRaheem, J. B. Awotunde, A. O. Balogun, R. M. Isiak, and T. O. Aro, "Empirical analysis of data streaming and batch learning models for network intrusion detection," *Electronics*, vol. 11, no. 19, p. 3109, Sep. 2022.
- [11] D. Bhosale and R. Ade, "Intrusion detection using incremental learning from streaming imbalanced data," *Int. J. Manag. Public Sector Inf. Commun. Technol.*, vol. 6, no. 1, pp. 9–20, Mar. 2015.
- [12] M. R. Mohamed, A. A. Nasr, I. F. Tarrad, and M. Z. Abdulmageed, "Exploiting incremental classifiers for the training of an adaptive intrusion detection model," *Int. J. Netw. Secur.*, vol. 21, no. 2, pp. 275–289, 2019.
- [13] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under concept drift: A review," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 12, pp. 2346–2363, Dec. 2018.
- [14] S. Wares, J. Isaacs, and E. Elyan, "Data stream mining: Methods and challenges for handling concept drift," *Social Netw. Appl. Sci.*, vol. 1, no. 11, pp. 1–19, Nov. 2019.
- [15] X. Yuan, R. Wang, Y. Zhuang, K. Zhu, and J. Hao, "A concept drift based ensemble incremental learning approach for intrusion detection," in *Proc. IEEE Int. Conf. Internet Things (iThings), IEEE Green Comput. Commun. (GreenCom), IEEE Cyber, Phys. Social Comput. (CPSCom), IEEE Smart Data (SmariData)*, Mar. 2018, pp. 350–357.
- [16] N. Littlestone and M. K. Warmuth, "The weighted majority algorithm," *Inf. Comput.*, vol. 108, no. 2, pp. 212–261, Feb. 1994. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0890540184710091>
- [17] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. ICISp*, vol. 1, 2018, pp. 108–116.
- [18] B. Alotaibi and K. Elleithy, "A majority voting technique for wireless intrusion detection systems," in *Proc. IEEE Long Island Syst., Appl. Technol. Conf. (LISAT)*, Apr. 2016, pp. 1–6.
- [19] M. Data and M. Aritsugi, "AB-HT: An ensemble incremental learning algorithm for network intrusion detection systems," in *Proc. Int. Conf. Data Sci. Appl. (ICoDSA)*, Jul. 2022, pp. 47–52.
- [20] A. Bifet and R. Gavaldà, "Learning from time-changing data with adaptive windowing," in *Proc. SIAM Int. Conf. Data Mining*, 2007, pp. 443–448.
- [21] A. A. Abuomman and M. B. I. Reaz, "A survey of intrusion detection systems based on ensemble and hybrid classifiers," *Comput. Secur.*, vol. 65, pp. 135–152, Mar. 2017.
- [22] B. A. Tama and S. Lim, "Ensemble learning for intrusion detection systems: A systematic mapping study and cross-benchmark evaluation," *Comput. Sci. Rev.*, vol. 39, Feb. 2021, Art. no. 100357. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1574013720304573>
- [23] M. Torabi, N. I. Udzir, M. T. Abdullah, and R. Yaakob, "A review on feature selection and ensemble techniques for intrusion detection system," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 5, pp. 6–7, 2021. [Online]. Available: <https://api.semanticscholar.org/CorpusID:236317529>
- [24] A. M. Bamhdi, I. Abrar, and F. Masoodi, "An ensemble based approach for effective intrusion detection using majority voting," *Telkonnika*, vol. 19, no. 2, pp. 664–671, Apr. 2021.
- [25] D. R. Patil and T. M. Patterwar, "Majority voting and feature selection based network intrusion detection system," *EAI Endorsed Trans. Scalable Inf. Syst.*, vol. 9, no. 6, p. e6, 2022.
- [26] H. Xu and Y. Wang, "A continual few-shot learning method via meta-learning for intrusion detection," in *Proc. IEEE 4th Int. Conf. Civil Aviation Saf. Inf. Technol. (ICCASIT)*, Oct. 2022, pp. 1188–1194.
- [27] T. Wang, Q. Lv, B. Hu, and D. Sun, "A few-shot class-incremental learning approach for intrusion detection," in *Proc. Int. Conf. Comput. Commun. Netw. (ICCCN)*, Jul. 2021, pp. 1–8.
- [28] J. Zheng, X. Ni, L. Li, K. Yu, and J. Zhang, "An ensemble learning-based two-level network intrusion detection method," in *Proc. Int. Conf. Comput. Eng. Artif. Intell. (ICCEAI)*, 2022, pp. 571–575.
- [29] C. Constantinides, S. Shiaeles, B. Ghita, and N. Kolokotronis, "A novel online incremental learning intrusion prevention system," in *Proc. 10th IFIP Int. Conf. New Technol., Mobility Secur. (NTMS)*, Jun. 2019, pp. 1–6.
- [30] S. Ndichu, T. Ban, T. Takahashi, and D. Inoue, "Critical-threat-alert detection using online machine learning," in *Proc. IEEE Int. Conf. Big Data*, Dec. 2022, pp. 3007–3014.
- [31] A. Bifet and R. Gavaldà, "Adaptive learning from evolving data streams," in *Proc. Int. Symp. Intell. Data Anal. (IDA)*. Cham, Switzerland: Springer, 2009, pp. 249–260.
- [32] B. Celik and J. Vanschoren, "Adaptation strategies for automated machine learning on evolving data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 9, pp. 3067–3078, Sep. 2021.
- [33] E. Fix and J. L. Hodges, "Discriminatory analysis: Nonparametric discrimination: Consistency properties," UASF School Aviation Med., Randolph AFB, TX, USA, Project 21-49-004, Tech. Rep. 4, 1951, pp. 261–279.
- [34] D. R. Cox, "The regression analysis of binary sequences," *J. Roy. Stat. Soc., Ser. B, Methodol.*, vol. 20, no. 2, pp. 215–232, Jul. 1958.
- [35] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.
- [36] River Development Team. (2021). *River: Online Machine Learning in Python*. [Online]. Available: <https://github.com/online-ml/river>
- [37] M. Data and M. Aritsugi, "T-DFNN: An incremental learning algorithm for intrusion detection systems," *IEEE Access*, vol. 9, pp. 154156–154171, 2021.
- [38] M. Tavallaei, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl.*, Jul. 2009, pp. 1–6.



ALIMOV ABDULBORIY received the B.S. degree in computer engineering from the Tashkent University of Information Technologies (named after Muhammad Al-Khwarizmi), Tashkent, Uzbekistan, in 2022. He is currently pursuing the Ph.D. degree with the Computer and Information Security Department, Sejong University, Seoul, South Korea. His research interests include information security, intrusion detection, and malware analysis.



Ji Sun Shin (Member, IEEE) received the B.S. degree in computer science from Seoul National University, Seoul, South Korea, in 2001, and the Ph.D. degree from the University of Maryland, College Park, USA, in 2009. From 2009 to 2012, she was a Senior Engineer with Samsung SDS, Seoul, where she was involved in the development of network access control systems. Since 2012, she has been an Associate Professor with the Computer and Information Security Department,

Sejong University. Her research interests include computer network security, cryptographic protocols, and applied cryptography.

• • •