Dual Adaptive Windows toward Concept-Drift in Online Network Intrusion Detection

Xiaowei Hu[1,2], Duohe Ma[1,2], Wen Wang[1,2(✉)], and Feng Liu[1,2]

[1] Institute of Information Engineering, CAS, Beijing, China
[2] School of Cyber Security, University of CAS, Beijing, China
{huxiaowei, maduohe, wangwen, liufeng}@iie.ac.cn

Abstract. Network intrusion detection is a commonly used and critical defense mechanism in the field of cybersecurity for identifying abnormal traffic online. However, the phenomenon of concept drift leads to a decrease in the accuracy of online intrusion detection systems in recognizing malicious traffic. Traditional machine learning-based intrusion detection systems are unable to adapt to the changes in data distribution of dynamic data streams. To address this issue, we propose DWOIDS, an online intrusion detection system based on dual adaptive windows and a Hoeffding tree classifier. When concept drift occurs in network data streams, it employs dual adaptive windows to monitor the prediction error of the classifier, continuously refining the classifier's accuracy in identifying malicious traffic. We conducted experimental evaluations on multiple datasets. Our proposed method demonstrated superior classification performance when compared to the state-of-the-art.

Keywords: Concept-Drift · Intrusion Detection · Adaptive Windows.

1 Introduction

In the increasingly interconnected digital ecosystem, the internet has facilitated a surge in networked devices, with network adversaries infiltrating these devices through data streams to commit data misuse. Consequently, the demand for robust online Network Intrusion Detection Systems (NIDS) has intensified. Online NIDS [1] are vital for maintaining security by continuously monitoring network traffic in real-time to detect malicious and suspicious activities. In recent years, research on Intrusion Detection Systems (IDS) has primarily been divided into two categories: rule-based IDS [8,25] and IDS based on Machine Learning (ML) [3,23] or Deep Learning (DL) [5,9,22]. Rule-based IDS relies on a predefined set of rules and identifies malicious features in network traffic through pattern matching. This approach offers the advantages of high detection accuracy and rapid detection speed. However, its static rules are unable to adapt to unknown threats and require regular updates. To overcome this limitation, researchers have recently begun to integrate ML and DL technologies to develop IDS with greater adaptability. Gao et al. [12] integrated various classifiers such as decision trees, random forests, KNN, and DNN, and dynamically adjusted their weights

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97626-1_15

to address declining classification accuracy. Chiche et al. [7] proposed a method combining machine learning and knowledge systems to enhance the adaptability and scalability of intrusion detection. These methods involve pre-training models and optimizing parameters using gradient descent to effectively fit threat traffic. Empirical studies have demonstrated that these methods possess a certain degree of effectiveness in detecting unknown threats. However, the presence of concept drift leads to a decline in the recognition accuracy of intrusion detection models trained on static datasets over time. While retraining these models with current data is a feasible solution to maintain their optimal performance, it incurs significant resources and time overhead.

The phenomenon of concept drift [18] refers to the non-stationarity of data stream distributions over time, which in the realm of cybersecurity manifests as unpredictable changes in the distribution of network data streams. Previous work typically assumes that data streams within a network originate from a single source and exhibit stable distributions. Consequently, when faced with dynamically shifting data distributions, the trained ML or DL intrusion detection models may fail to adapt to these changes, leading to a degradation in their predictive performance. Although the research on concept drift in IDS [20] has garnered attention, it remains in its infancy. The prevalent approaches involve directly transferring or integrating traditional concept drift detection techniques. For instance, the ADWIN [6] algorithm maintains a variable sliding window to statistically analyze the data distribution changes between two sub-windows, using confidence levels to ascertain whether the distribution change exceeds a predefined threshold. Similarly, the DDM [11] method detects concept drift by calculating whether the model's prediction error rate surpasses a threshold. Meanwhile, HDDM [10] computes the moving average of the model's predictions and employs the Hoeffding inequality to determine the threshold boundary for drift.

The challenges faced by existing IDS can be summarized into four key points:

- Traditional IDS, trained on fixed rules or static datasets, are vulnerable to future unknown attack patterns.
- The data distribution of online network traffic evolves over time, and ML or DL models trained on static datasets may struggle to adapt to such dynamic changes.
- The cost of retraining models to replace outdated ones in the event of concept drift is excessively high, posing practical challenges.
- Traditional concept drift detection methods have not been specifically optimized for online network traffic, thereby limiting their detection performance to some extent.

To overcome these challenges, we propose DWOIDS, an online intrusion detection system framework based on dual adaptive windows and the Hoeffding Tree classifier. This framework utilizes the Hoeffding Tree as the base classifier for real-time prediction of malicious behaviors in online network traffic. To effectively mitigate concept drift, we innovatively calculate the momentum of prediction errors within fast and slow windows, and estimate the degree of concept

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97626-1_15

drift through the moving average difference of these momenta. Upon detection of concept drift, the framework automatically replaces the base classifier with a new classifier learned under the context of concept drift, thereby significantly reducing the overhead of model replacement and enhancing the framework's adaptive capabilities and detection efficiency.

The main contributions of this paper are as follows:

- We propose an online intrusion detection framework based on dual adaptive windows and the Hoeffding tree classifier. This framework effectively addresses the issue of concept drift in real-time network flows, thereby significantly improving the adaptability and detection efficiency of the intrusion detection system.
- To accurately identify concept drift in network flows, we design the DAWMA algorithm. This algorithm utilizes momentum and its moving average difference as estimators, effectively enhancing the intrusion detection performance by recognizing concept drift.
- We comprehensively evaluate our proposed framework on two public datasets, CIC-IDS2017[21] and NSL-KDD[24], and compare it with state-of-the-art methods. The experimental results demonstrate that our framework excels in handling intrusion detection tasks with concept drift.

The structure of this paper is organized as follows: Section 2 systematically reviews the relevant research on concept drift in the field of IDS. Section 3 provides a detailed presentation of the innovative framework we propose. Section 4 discusses the experiments on the proposed framework to validate its effectiveness and superiority. Finally, Section 5 offers a comprehensive summary of the entire paper.

2 Related Work

In the context of online IDS, the application of concept drift is generally approached in two ways. One method involves incorporating the incoming network data stream into an adaptive window and then statistically analyzing the distributional changes between two sub-windows. If the change exceeds a predefined threshold, it is deemed that concept drift has occurred. Another method involves employing a classifier to initially predict network data streams and determine their real-time accuracy. Following this, the moving average of the accuracy or other statistical measures is evaluated. When the evaluated changes exceed a certain threshold, it is concluded that concept drift has taken place.

2.1 Data Distribution-Based Concept Drift Mitigation in IDS

Yang et al. [27] introduced the CADE system for detecting and explaining concept drift samples in security applications. CADE maps training data to a low-dimensional space using contrastive learning and learns a distance function between samples to identify drift samples deviating from the training distribution.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97626-1_15

However, It struggles with sparse data, high dimensionality, and limited performance on in-class evolution scenarios. Jain et al. [16] used a sliding window and employed K-Means clustering to reduce the data volume and update the training set. Subsequently, KL divergence is employed to measure data distribution differences and an SVM classifier is used for anomaly detection, and the model is retrained based on statistical tests. It achieves high accuracy but depends on manual parameter tuning and may not scale well with large or real-time data streams. Rajeswari et al. [19] introduced an efficient intrusion detection system based on concept drift in data streams and Support Vector Machines (SVM). The system enhances data quality through data cleansing and normalization techniques and facilitates pattern extraction and data change detection by incorporating a timestamp attribute for data streaming. The ADWIN algorithm is utilized to detect concept drift in the data stream, splitting the window into two parts at the cut-off point. An SVM classifier is then employed to categorize the data into normal and attack classes.

2.2 Prediction Statistics-Based Concept Drift Mitigation in IDS

Andresini et al. [2] introduced INSOMNIA, a semi-supervised intrusion detection framework designed to address the issue of concept drift in network traffic features over time. INSOMNIA employs a DNN classifier integrated with active learning, label estimation, and XAI to enhance adaptability. Drift is detected by assessing DNN accuracy against pseudo-labels from an NC classifier, triggering model updates via the NC classifier to align with new traffic patterns. While the approach demonstrates robustness to concept drift, it struggles with detecting low-prevalence attacks and requires further work to generalize across diverse attack types. Yang et al. [26] addressed the issue of concept drift in Internet of Things (IoT) data streams by proposing an integrated framework named PWPAE. PWPAE combines two popular drift detection methods (ADWIN and DDM) with two state-of-the-art drift adaptation methods (ARF and SRP) to construct base learners. Subsequently, the PWPAE is used to weight the base learners based on their real-time performance, and to integrate them into a robust anomaly detection ensemble model, thereby enhancing the drift adaptation performance. While PWPAE outperforms existing approaches in drift handling, broader applicability and efficiency in resource-constrained settings require further study.

Jain and Kaur [15] introduced a hybrid ensemble technique based on concept drift detection for distributed anomaly detection in network data streams. This technique integrates random forests and logistic regression as the first-level classifiers, with support vector machines serving as the second-level classifiers. To address concept drift, they employed a sliding window-based K-Means clustering technique to reduce the data volume and update the training set. Hnamte and Hussain [14] proposed a hybrid deep learning model named DCNNBiLSTM for network intrusion detection. This model leverages the strengths of both Convolutional Neural Networks (CNN) and Bidirectional Long Short-Term Memory networks (BiLSTM), and is optimized through a Deep Neural Network (DNN).

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97626-1_15

The DCNNBiLSTM model first employs CNN to extract local features from the input data, followed by the use of BiLSTM to capture long and short-term dependencies in sequential data, with the final classification being performed by the DNN. However, the proposed model's high parameter complexity might exhibit overfitting tendencies. Seth et al. [20] proposed an intrusion detection method based on Adaptive Random Forest (ARF). ARF adapts in real-time to the evolving network environment and attack patterns, and prioritizes new data through an instance weighting mechanism, thereby enhancing the intrusion detection capability. Hoeffding's inequality and the moving average test provide statistical support to the system, aiding in the timely identification of concept drift and distinguishing between benign network changes and potential intrusions. However, Computational overhead from drift detection and adaptive updates may hinder scalability.

3 Methodology

3.1 Overall Architecture

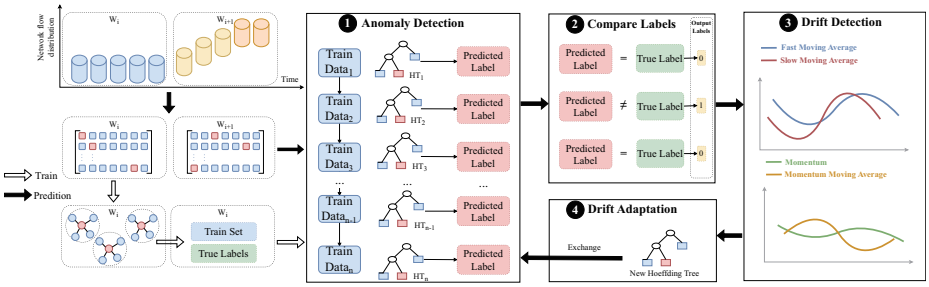


Fig. 1. Overview of DWOIDS

Our proposed DWOIDS uses a dual adaptive window moving average technique (DAWMA) to detect concept drift and employs Hoeffding Trees as the base classifier. As shown in the Fig. 1, this method first learns the characteristics of network traffic in the W_i window, and then performs predictions in the W_{i+1} window. The proposed framework initially preprocesses the data stream through one-hot encoding, min-max normalization, and Birch [28] clustering to extract representative training data streams. Following this, it employs Hoeffding trees to learn and predict network traffic in the current environment, thereby generating corresponding prediction labels. The binary prediction results are then added to both a fast and a slow window, with the difference between the mean values of these windows being termed as the dual window mean. Subsequently, we introduce an estimator based on the difference between the momentum moving averages of fast and slow windows. Furthermore, we trigger a drift signal if the

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97626-1_15

sign of the estimator at the current time step is opposite to that of the preceding time step. Lastly, the classifier that was learned in the context of the detected concept drift replaces the base classifier. This section offers a comprehensive description of the modules within the framework.

3.2 Data Processing

Due to the vast scale of network traffic data streams and in order to maintain the performance of the online IDS, we employ BIRCH clustering for the sampling of training data. We chose the BIRCH clustering algorithm because it automatically identifies high-quality cluster centers without pre-setting the number of clusters, making it highly effective for large-scale network data streams.

Algorithm 1: Birch Clustering Algorithm

input : Network data stream setX $\{x_i, ...x_j\}$, label setY $\{y_i, ...y_j\}$, threshold T, branching factor B
output: The final retained clusters

Apply One-Hot encoding to the input sets;
Perform min-max normalization on the encoded data;
Initialize an empty CF Tree;
for $i \leftarrow 1$ **to** *length*(*setX*) **do**
    Find the closest leaf node L in the CF Tree to x_i;
    if *after adding* x_i *to node L, the diameter of* $L < T$ **then**
        adding x_i to L
    if *the number of CFs in* $L < B$ **then**
        Create a new CF for x_i in L
    else split L to L' and adding x_i to node L';
if *the number of CFs in the root of the CF Tree* $\geq B$ **then**
    compress the tree
ClusterLabels $\leftarrow$ *global clustering on CF Tree*;
clusters $\leftarrow$ *mapping of setX, setY and ClusterLabels*;
nc $\leftarrow$ *number of clusters*;
for $i \leftarrow 1$ **to** *nc* **do**
    if *the ratio of malicious labels in the* $cluster_i < 0.2$ **then**
        discard $cluster_i$
return clusters

In this paper, the network data stream is represented as setX $\{x_i, x_{i+1}, ..., x_j\}$, with i and j indicating different time points. And the label setY $\{y_i, y_{i+1}, ..., y_j\}$ comprises two states: 'normal' and 'malicious'. Specifically, we initially perform one-hot encoding on the data stream features to facilitate subsequent calculations. Afterwards, we apply min-max scaling to the encoded results to ensure

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97626-1_15

data consistency and comparability. Algorithm 1 presents the specific steps of BIRCH clustering. Through the precise application of the BIRCH clustering algorithm, we successfully delineated multiple clusters, each composed of similar data streams.

3.3 Detecting Concept Drift with Dual Adaptive Windows

We input the processed network data stream into the classifier for prediction, with details of the classifier provided in Section 3.4. The classifier outputs a set of predicted labels $\{p_i, ...p_j\}$. We introduce the prediction error (pe) as an quantify metric of classifier. As illustrated in Equation 1, the prediction error is defined as follows: it takes the value of 0 when the predicted label matches the actual label. Conversely, the prediction error is set to 1.

$$pe_i = \begin{cases} 0, & p_i = y_i \\ 1, & p_i \neq y_i \end{cases} \tag{1}$$

To mitigate the limitations of relying solely on single-point prediction errors for detecting concept drift, we introduce the Dual-Adaptive Window Moving Average (DAWMA) algorithm. This approach aims to balance the sensitivity and accuracy of drift detection. The DAWMA algorithm employs a dual-sliding window mechanism, comprising a fast window and a slow window, both of which continuously receive the most recent pe_i values. Initially, the slow window leads the fast window by d units in length. As data continues to be input, the lengths of both windows gradually increase until concept drift is detected. During this process, we denote the lengths of the fast and slow windows as fl and sl, respectively, and their respective average values as fa and sa. The dual-window mechanism effectively balances the needs of short-term and long-term strategies. Furthermore, the windows in this algorithm are designed to dynamically adapt: if no concept drift is detected over a long period, the window size will automatically increase to capture broader data trends; conversely, if concept drift occurs frequently in the short term, the window will maintain a smaller size to enhance sensitivity to rapid changes.

$$M = (fa_k - sa_k) - \frac{\sum_{i=k-n}^{k-1} (fa_i - sa_i)}{n} \tag{2}$$

As illustrated in the Formula 2, the DAWMA algorithm introduces the momentum of the slow window and the fast window to quantify the intensity of changes in the prediction error. Subsequently, the moving average difference M of these momenta is calculated to effectively filter noise and short-term data fluctuations. Furthermore, if the sign of the current M is opposite to that of the previous M, it indicates a trend of strengthening or weakening in the momentum of the prediction error, at which point we determine that a concept drift has occurred.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97626-1_15

3.4 Classifier Based on Hoeffding Tree

Hoeffding Tree is an incremental decision tree algorithm for data stream classification. It processes large-scale data streams in real-time, learning and predicting on the fly without storing all historical data. In this study, we assume that the network data stream $\{x_i, x_{i+1}, ..., x_j\}$ is mutually independent and bounded $(a_i \leq x_i \leq b_i)$. Based on this assumption, for any constant $\theta > 0$, the expression represented by Equation 3 conforms to the Hoeffding's bound.

$$P\left(\sum_{i=1}^{n} x_i - E\left[\sum_{i=1}^{n} x_i\right] \geq \theta\right) \leq \exp\left(-\frac{2\theta^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right) \tag{3}$$

This study constructs a decision tree model based on Hoeffding's inequality. Initially, the Hoeffding Tree consists of a single root node, which is responsible for storing the class distribution information of the initial data. As new network data streams arrive, these data start from the root node and are directed to the corresponding child nodes based on the splitting attributes and feature values of each node. This process is recursive and continues until the data reach a leaf node. Upon arrival at a leaf node, the node updates its class distribution information to reflect the incorporation of new data. During the splitting decision of the decision tree, we use information gain as a metric, which is defined as the difference between entropy and conditional entropy. Then, we utilize Hoeffding's inequality to determine whether the attribute with the currently observed maximum information gain exceeds the Hoeffding bound, thereby deciding whether to perform node splitting. Since Hoeffding Tree is an incremental learning algorithm, once concept drift is detected, new concepts are gradually integrated into the Hoeffding Tree, thereby demonstrating the algorithm's robustness.

4 Experiments Evaluation

This section delves into the experimental evaluation process, wherein we utilized the CIC-IDS2017 [21] and NSL-KDD [24] datasets for testing. Under the condition of consistent dataset distribution and environmental setup, we reproduced and conducted a comparative analysis with other research works, thereby robustly validating the effectiveness of our proposed method.

4.1 Dataset and Processing

The CIC-IDS2017 dataset, published by the Canadian Institute for Cybersecurity, captures real network traffic from July 3, 2017, to July 7, 2017, and includes 15 common attack scenarios such as brute force FTP, brute force SSH, DoS, Heartbleed, web attacks, intrusions, Botnet, and DDoS. To enhance data quality, we utilized the improved version of the dataset by Liu et al. [17], which rectifies label errors and feature extraction inaccuracies in the original dataset and removes meaningless artefacts.Ultimately, we acquired a total of 2,090,564 network traces, comprising 1,657,069 benign traces and 433,495 malicious traces.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97626-1_15

The specific distribution of the malicious traces is presented in Table 1. Due to the significant imbalance between benign and malicious tracks, it is necessary to consider the base-rate fallacy [4].

Table 1. Statistical Analysis of Malicious Traces

Attack Types	Count	Proportion
Infiltration	38	0.0088%
Brute Force	151	0.0348%
SQL Injection	12	0.0028%
XSS	27	0.0062%
DoS GoldenEye	7,567	1.7456%
DoS Hulk	158,469	36.5561%
DoS Slowhttptest	1,742	0.4019%
DoS Slowlori	4,001	0.9230%
Heartbleed	11	0.0025%
FTP Patator	3,973	0.9165%
SSH Patator	2,980	0.6874%
Botnet	738	0.1702%
DDoS	94,763	21.8602%
Portscan	159,023	36.6839%
SUM	433,495	100.0000%

The NSL-KDD dataset is an enhanced version of the KDD Cup 1999 dataset, specifically designed for the evaluation of network intrusion detection systems. It comprises 148,517 records, each consisting of 41 traffic features and a classification label. The dataset encompasses five distinct types of data, namely Normal (normal traffic), DOS (Denial of Service attacks), Probe (probing attacks), R2L (Remote to Local attacks), and U2R (User to Root attacks). Among these, benign traffic constitutes 53% of the dataset, while malicious traffic accounts for the remaining 47%. The dataset contains 125,973 training samples and 22,544 testing samples.

4.2 Experimental Setup

Our experiments were conducted in an environment featuring Ubuntu 22.04 as the operating system, an x86 hardware architecture, a 32-core AMD CPU, and 1TB of RAM. We implemented the proposed framework in Python and set the BIRCH clustering threshold to 0.5. In addition, to ensure a fair comparison, we obtained the source code for all baseline works and keep the best configurations

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97626-1_15

for all hyperparameters. Furthermore, we conducted experiments using the same dataset distribution and proportions to replicate their results.

In our experiments with the CIC-IDS2017 dataset, we adopted the same approach as previous work [2,16] by using windows containing 50,000 instances to highlight the distribution characteristics of concept drift. For the NSL-KDD training set containing 125,973 samples, we configured the window size as 7,000, while for the test set with 22,544 samples, a window size of 2,000 was implemented. However, unlike prior studies, we discard the last window if it's smaller than the set size. Specifically, we trained a model in the current Window Id W_i (i starts from 0) and then used this trained classifier to predict in the subsequent Window Id W_{i+1}.

4.3 Evaluation Metrics

In this experiment, we employ accuracy, false positive rate (FPR), precision, recall, and F1 score as the metrics for performance evaluation. Specifically, true positive instances are denoted as (TP), false positive instances as (FP), true negative instances as (TN), and false negative instances as (FN). The mathematical expressions for these evaluation metrics are presented in Equations 4-8.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{4}$$

$$\text{FPR} = \frac{FP}{FP + FN} \tag{5}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{6}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{7}$$

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

4.4 Evaluation Results and Discussions

The aim of this study is to comprehensively evaluate the performance of our proposed DWOIDS in network intrusion detection with concept drift. To this end, we have designed the following two experiments:

- Classification Experiment: In this experiment, we do not apply special treatment to concept drift, aiming to evaluate the prediction accuracy of the base classifier (based on the Hoeffding tree) and address the issue of classifier prediction bias.
- Intrusion Detection with Concept Drift Experiment: We will conduct experiments using our proposed comprehensive framework, and compare it with baseline methods to demonstrate the superior performance of our proposed framework.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97626-1_15

Table 2. The Window where the Classifier Fails

Dataset	Classifier	Failed Window Id	Failed Count
CIC-IDS2017	only Hoeffding tree	$W_1, W_6, W_7, W_{12}, W_{14}, W_{15}, W_{17}, W_{18}, W_{19}, W_{22}, W_{23}, W_{29}, W_{32}, W_{34}, W_{35}, W_{36}, W_{37}, W_{38}$	18
	add DAWMA	$W_{14}, W_{18}, W_{35}, W_{36}, W_{38}$	5
NSL-KDD	only Hoeffding tree	$W_3, W_4, W_6, W_{11}, W_{12}$	5
	add DAWMA	/	0

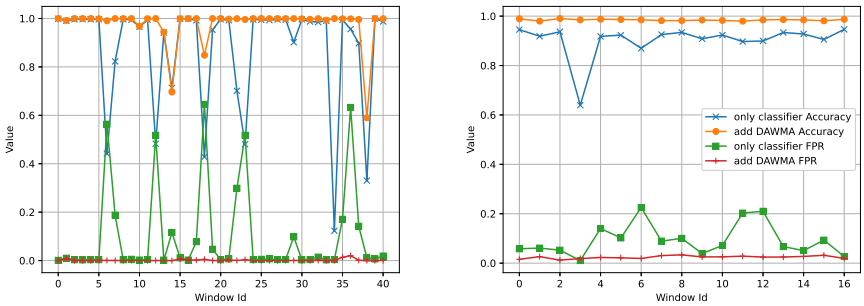
Classification Experiment This study assesses the impact of concept drift handling on prediction performance using the CIC-IDS2017 and NSL-KDD datasets. The 2,090,564 CIC-IDS2017 samples are divided into 41 windows, while the 125,973 NSL-KDD samples are split into 17 windows. The experiment comprised two phases: In the first phase, the base Hoeffding tree classifier was applied to predict data streams within each window, with performance evaluated through accuracy and false positive rate (FPR). The second phase introduced the Dual-Adaptive Window Moving Average (DAWMA) method to address concept drift, followed by recalculation of these metrics post-optimization. The experimental results are presented in Fig 2. Then, we compute the average "add DAWMA Accuracy" (average Acc) and "add DAWMA FPR" (average FPR). We establish a failure criterion: The base classifier is deemed to have failed in addressing concept drift if either the accuracy metric (acc_i) in window (W_i) satisfies the threshold condition defined in Equation 9, or the false positive rate (fpr_i) reaches the critical value specified by Equation 10. Table 2 presents the Window Ids where the base classifier fails before and after adding DAWMA. Empirical analysis reveals a substantial decrease in the number of windows experiencing base classifier failure due to concept drift following the incorporation of DAWMA, thereby enhancing the classifier's robustness.

$$acc_i < \text{average Acc} \times 0.9 \tag{9}$$

$$fpr_i \times 0.2 > \text{average FPR} \tag{10}$$

This experiment highlights three crucial findings: Firstly, concept drift indeed exists in network data streams, posing a significant challenge for traditional classifiers which often result in classifier failure. Secondly, the introduction of our proposed DAWMA strategy enables rapid recovery and stable performance of the classifier, thereby underscoring the necessity of accounting for concept drift in the context of network stream online intrusion detection. Thirdly, in the event of concept drift in the current window, our proposed framework swiftly replaces the outdated classifier with a newly trained one, which then effectively monitors the subsequent window. Experimental results demonstrate the effectiveness of our approach.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97626-1_15



(a)CIC-IDS2017 (b)NSL-KDD

Fig. 2. CIC-IDS2017 and NSL-KDD in Concept Drift Handling

Intrusion Detection with Concept Drift Experiment In this study, We reproduce all baseline methods and perform detailed comparative analysis. The selected baseline methods are categorized into two groups: those that do not deal with concept drift, like Naive Bayes and LightGBM, and those that do, such as ARF+ADWIN[13], INSOMNIA [2].

For the CIC-IDS2017 dataset, we selected 1,396,914 traces from the last three days for validation with a window size of 50,000. For the NSL-KDD dataset, we used 22,544 test set traces for the experiment, with a window size of 2,000. The experiment was designed to train models in the current window and use the trained models to make predictions in the next window. Ultimately, we compared the accuracy, precision, recall, and F1 score of each model.

Specifically for the INSOMNIA method, which requires pre-training a deep neural network (DNN) model and making predictions on traces within the test window during testing, while also using fine-tuning strategies to adjust model performance in real-time, we used 693,650 traces from the first two days of CIC-IDS2017 and 125,973 training set traces from NSL-KDD to train the model for evaluation. Subsequently, we validated the model performance using the same approach as other methods and conducted a comparative analysis within the same test window interval to ensure the fairness of the experiment and the comparability of the results.

Table 3 shows the average results of all experiments conducted on the CIC-IDS2017 and NSL-KDD datasets. Our analysis reveals that methods accounting for concept drift significantly outperform those that do not when detecting real-time data streams. This is attributed to the former to adjust their original prediction strategies, either through fine-tuning or model replacement, following the detection of concept drift. This observation supports our contention that the distribution of network data streams evolves over time, and traditional static Intrusion Detection Systems (IDS) are less capable of adapting to concept drift. Notably, INSOMNIA encounters difficulties in detecting low-frequency and covert attacks (such as Infiltration attacks), and its update mechanism fails to

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97626-1_15

Table 3. Comparison of Algorithms for Online Intrusions Detection

Method	Dataset	Accuracy	Precision	Recall	F1 Score
Naive Bayes*	NSL-KDD	0.8563	0.8199	0.8730	0.8345
	CIC-IDS2017	0.8787	0.9212	0.9067	0.8758
LightGBM*	NSL-KDD	0.9418	**0.9829**	0.8802	0.9284
	CIC-IDS2017	0.9401	0.8657	0.8020	0.7982
ARF+ADWIN	NSL-KDD	0.9340	0.9426	0.9028	0.9210
	CIC-IDS2017	0.9383	0.9362	**0.9972**	0.9532
INSOMNIA	NSL-KDD	0.5455	0.4666	0.8178	0.5940
	CIC-IDS2017	0.8656	0.3941	0.4857	0.3758
We Proposed DWOIDS	NSL-KDD	**0.9463**	0.9524	**0.9214**	**0.9364**
	CIC-IDS2017	**0.9608**	**0.9480**	0.9969	**0.9653**

* indicates that this method does not address concept drift.

effectively utilize the knowledge of attacks that occur only within a single window (such as Brute Force attacks), resulting in degraded detection performance in certain windows and thereby affecting the overall average performance metrics. Furthermore, our proposed framework achieved the best performance in accuracy, recall, and F1 score on the NSL-KDD dataset. Similarly, on the CIC-IDS2017 dataset, the framework also attained optimal results in accuracy, precision, and F1 score. Experimental results demonstrate that our proposed method not only outperforms traditional machine learning and deep learning algorithms but also exhibits significant advantages over other algorithms in handling concept drift in intrusion detection.

5 Conclusion and Future Work

This paper presents DWOIDS, an innovative online network intrusion detection system framework based on dual adaptive windows, designed to effectively address the challenge of concept drift in the field of network intrusion detection. This framework relies on a base classifier based on the Hoeffding tree to accurately identify malicious components in network traffic. Experiments on NSL-KDD and CIC-IDS2017 datasets demonstrate the necessity of addressing concept drift in online intrusion detection, and our proposed framework outperforms the baseline model. This enables real-time adaptation in critical scenarios, where concept drift frequently disrupts anomaly detection.

Currently, our validation is limited to the binary classification problem of online network streams using benchmark datasets. To bridge the academia-industry gap, we plan to deploy DWOIDS in real-world environments in future work, focusing on evaluating its robustness to multi-class concept drift against emerging network attack traffic.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97626-1_15

Acknowledgement. This paper was supported by National Natural Science Foundation of China(No.62472418).

References

1. Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., Ahmad, F.: Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Transactions on Emerging Telecommunications Technologies **32**(1), e4150 (2021)
2. Andresini, G., Pendlebury, F., Pierazzi, F., Loglisci, C., Appice, A., Cavallaro, L.: Insomnia: Towards concept-drift robustness in network intrusion detection. In: Proceedings of the 14th ACM workshop on artificial intelligence and security. pp. 111–122 (2021)
3. Arp, D., Quiring, E., Pendlebury, F., Warnecke, A., Pierazzi, F., Wressnegger, C., Cavallaro, L., Rieck, K.: Dos and don'ts of machine learning in computer security. In: 31st USENIX Security Symposium (USENIX Security 22). pp. 3971–3988 (2022)
4. Axelsson, S.: The base-rate fallacy and the difficulty of intrusion detection. ACM Transactions on Information and System Security (TISSEC) **3**(3), 186–205 (2000)
5. Bao, H., Li, W., Wang, X., Tang, Z., Wang, Q., Wang, W., Liu, F.: Payload level graph attention network for web attack traffic detection. In: International Conference on Computational Science. pp. 394–407. Springer (2023)
6. Bifet, A., Gavalda, R.: Learning from time-changing data with adaptive windowing. In: Proceedings of the 2007 SIAM international conference on data mining. pp. 443–448. SIAM (2007)
7. Chiche, A., Meshesha, M.: Towards a scalable and adaptive learning approach for network intrusion detection. Journal of Computer Networks and Communications **2021**(1), 8845540 (2021)
8. Feng, X., Liao, X., Wang, X., Wang, H., Li, Q., Yang, K., Zhu, H., Sun, L.: Understanding and securing device vulnerabilities through automated bug report analysis. In: SEC'19: Proceedings of the 28th USENIX Conference on Security Symposium (2019)
9. Ferrag, M.A., Maglaras, L., Moschoyiannis, S., Janicke, H.: Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. Journal of Information Security and Applications **50**, 102419 (2020)
10. Frias-Blanco, I., del Campo-Ávila, J., Ramos-Jimenez, G., Morales-Bueno, R., Ortiz-Diaz, A., Caballero-Mota, Y.: Online and non-parametric drift detection methods based on hoeffding's bounds. IEEE Transactions on Knowledge and Data Engineering **27**(3), 810–823 (2014)
11. Gama, J., Medas, P., Castillo, G., Rodrigues, P.: Learning with drift detection. In: Advances in Artificial Intelligence–SBIA 2004: 17th Brazilian Symposium on Artificial Intelligence, Sao Luis, Maranhao, Brazil, September 29-Ocotber 1, 2004. Proceedings 17. pp. 286–295. Springer (2004)
12. Gao, X., Shan, C., Hu, C., Niu, Z., Liu, Z.: An adaptive ensemble machine learning model for intrusion detection. Ieee Access **7**, 82512–82521 (2019)
13. Gomes, H.M., Bifet, A., Read, J., Barddal, J.P., Enembreck, F., Pfharinger, B., Holmes, G., Abdessalem, T.: Adaptive random forests for evolving data stream classification. Machine Learning **106**, 1469–1495 (2017)

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97626-1_15

14. Hnamte, V., Hussain, J.: Dcnnbilstm: An efficient hybrid deep learning-based intrusion detection system. Telematics and Informatics Reports **10**, 100053 (2023)
15. Jain, M., Kaur, G.: Distributed anomaly detection using concept drift detection based hybrid ensemble techniques in streamed network data. Cluster Computing **24**(3), 2099–2114 (2021)
16. Jain, M., Kaur, G., Saxena, V.: A k-means clustering and svm based hybrid concept drift detection technique for network anomaly detection. Expert Systems with Applications **193**, 116510 (2022)
17. Liu, L., Engelen, G., Lynar, T., Essam, D., Joosen, W.: Error prevalence in nids datasets: A case study on cic-ids-2017 and cse-cic-ids-2018. In: 2022 IEEE Conference on Communications and Network Security (CNS). pp. 254–262. IEEE (2022)
18. Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., Zhang, G.: Learning under concept drift: A review. IEEE transactions on knowledge and data engineering **31**(12), 2346–2363 (2018)
19. Rajeswari, P.V.N., Shashi, M., Rao, T.K., Rajya Lakshmi, M., Kiran, L.V.: Effective intrusion detection system using concept drifting data stream and support vector machine. Concurrency and Computation: Practice and Experience **34**(21), e7118 (2022)
20. Seth, S., Chahal, K.K., Singh, G.: Concept drift–based intrusion detection for evolving data stream classification in ids: Approaches and comparative study. The Computer Journal p. bxae023 (2024)
21. Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A., et al.: Toward generating a new intrusion detection dataset and intrusion traffic characterization. ICISSp **1**, 108–116 (2018)
22. Shen, Y., Mariconti, E., Vervier, P.A., Stringhini, G.: Tiresias: Predicting security events through deep learning. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. pp. 592–605 (2018)
23. Sommer, R., Paxson, V.: Outside the closed world: On using machine learning for network intrusion detection. In: 2010 IEEE symposium on security and privacy. pp. 305–316. IEEE (2010)
24. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A.: A detailed analysis of the kdd cup 99 data set. In: 2009 IEEE symposium on computational intelligence for security and defense applications. pp. 1–6. Ieee (2009)
25. Wang, X., Bao, H., Li, W., Chen, H., Wang, W., Liu, F.: A framework for intelligent generation of intrusion detection rules based on grad-cam. In: International Conference on Computational Science. pp. 147–161. Springer (2024)
26. Yang, L., Manias, D.M., Shami, A.: Pwpae: An ensemble framework for concept drift adaptation in iot data streams. In: 2021 IEEE Global Communications Conference (GLOBECOM). pp. 01–06. IEEE (2021)
27. Yang, L., Guo, W., Hao, Q., Ciptadi, A., Ahmadzadeh, A., Xing, X., Wang, G.: {CADE}: Detecting and explaining concept drift samples for security applications. In: 30th USENIX Security Symposium (USENIX Security 21). pp. 2327–2344 (2021)
28. Zhang, T., Ramakrishnan, R., Livny, M.: Birch: an efficient data clustering method for very large databases. ACM sigmod record **25**(2), 103–114 (1996)

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97626-1_15