ISSN 2007-9737

Adaptive Intrusion Detection System: Hybrid K-Means and Random Forest Approach with Concept Drift Detection

Eric García-Huitzitl[1,*], Lázaro Bustio-Martínez[2], René Cumplido[1]

[1] Instituto Nacional de Astrofísica, Óptica y Electrónica, Puebla, Mexico

[2] Universidad Iberoamericana Ciudad de México, Departamento de Estudios en Ingeniería para la Innovación, Mexico

{gahueric, rcumplido}@inaoep.mx, lazaro.bustio@ibero.mx

Abstract. The ever-evolving data landscape presents significant challenges, such as concept drift, where shifts in statistical distributions within data streams pose critical cybersecurity threats. Traditional machine learning, which relies on static models, struggles with concept drift, underscoring the necessity for adaptive approaches specifically designed for streaming data. This paper investigates methodologies aimed at enhancing security in dynamic data environments. A hybrid concept drift detection method that combines error rate analysis with data distribution monitoring is proposed. Additionally, to update the training dataset, the approach employs a combination of sliding window-based data capture and drift analysis, along with K-Means clustering and a Random Forest classifier. This includes the use of two types of sliding windows: fixed and adaptive. Adaptive Random Forest classifier is used to anomaly detection and retraining the model. Experiments were conducted on the NSL-KDD dataset to detect and quantify the severity of concept drift, utilizing techniques such as Principal Component Analysis and Spearman's Correlation Coefficient. Consequently, the performance of the Intrusion Detection System to adapt to these changes was also evaluated. The proposed adaptive model demonstrates significant enhancements, with Adaptive Random Forest achieving a classification accuracy of 98.66%. Furthermore, precision, detection rate, and F1-score rates of 99.52%, 97.74%, and 99.78%, respectively, are achieved. All this while maintaining a low false alarm rate of 1.14%.

Keywords. Adaptive IDS, concept drift, hybrid approach, clustering, classification.

1 Introduction

In today's interconnected world, cybersecurity is crucial for protecting digital infrastructure and sensitive information. In 2023, Kaspersky [10] reported detecting approximately 411,000 malicious files, marking an increase of nearly 3% compared to the previous year.

Among the identified malicious files were DDoS botnets, ransomware, miners, DNS changers, and proxy bots, which are notorious for compromising computer systems.

Furthermore, Kaspersky's honeypots revealed that 97.91% of brute-force password attempts were focused on Telnet, with only 2.09% targeting SSH. Therefore, to effectively navigate the continually evolving cybersecurity landscape and mitigate emerging threats, it is essential to implement advanced security measures [19].

In addressing this need, Intrusion Detection Systems (IDS) stand out as fundamental tools. In particular, anomaly-based IDS employing machine learning models actively monitor network traffic for abnormal activities or suspicious behavior patterns indicative of potential intrusion attempts or cyberattacks [3].

Upon detecting unusual activity, the IDS promptly generates alerts, enabling network administrators to swiftly implement corrective measures [16].

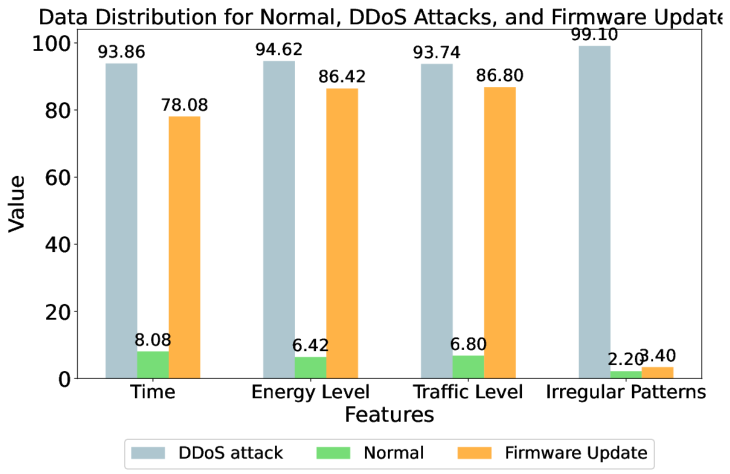


Fig. 1. Data distribution for normal, DDoS attacks, and firmware updates

This proactive approach is crucial for early threat detection, enhancing network security, and protecting an organization's digital assets.

However, the phenomenon of concept drift, which involves shifts in the statistical properties of input features and/or the target variable over time, can significantly impact predictive model performance by altering the relationship between input features and the target variable, potentially leading to decreased accuracy [12, 19].

Traditional IDS relying on static models, where a predictive model is established to forecast new instances, face significant challenges in effectively mitigating emerging cyber threats and attacks due to the evolving patterns and behaviors that IDS are designed to detect [16].

This phenomenon can be illustrated in Fig. 1, which shows the typical learned distribution of an IDS for detecting Distributed Denial-of-Service (DDoS) attacks [17]. Fig. 1 reveals fluctuations in time, energy, traffic level, and irregular patterns, characteristics typically associated with attacks.

However, this distribution may not always remain constant. For instance, a firmware update could alter the data distribution, causing it to resemble that of a DDoS attack and potentially leading to misclassification. This decrease in detection accuracy compels network administrators to explore alternative tools and solutions.

Therefore, it is crucial to develop adaptive models capable of adjusting to emerging patterns and variations in dynamic stream data [16]. In this context, incremental learning algorithms, which continuously update models with new data without requiring complete retraining, along with concept drift detection methods, are essential for maintaining model performance [19].

Additionally, models that do not rely exclusively on labeled datasets are needed, ensuring effective detection even in environments where labeled data availability is limited or nonexistent [9]. This approach ensures that IDS can maintain their sensitivity to the ever-changing cybersecurity landscape.

This paper aims, as its major contribution, to introduce an adaptive model for IDS that incorporates incremental learning through a hybrid unsupervised approach and a concept drift detection method based on distribution and error analysis.

Additionally, it presents an adaptive sliding window method for retraining the model, allowing it to accommodate emerging concepts or changes in data distribution. This approach autonomously identifies shifts in data patterns, enhancing the system's ability to respond to emerging threats without relying on predefined knowledge. The remainder of this paper is structured as follows.

Algorithm 1 Concept drift manager

```
Input:  Testing Data,
        Threshold  : $\mathrm{Thresh\_DR}$,
        Threshold  : $\mathrm{Thresh\_FAR}$
Output: Update adaptive random forest model.
 1: procedure $\mathrm{Adaptive\_Sliding\_Window(Testing\ data)}$
 2:     Divide the testing data into fixed sliding windows of size $n$, forming a set $S = \{SW_1,\ SW_2,\ SW_3,\ \cdots,\ SW_n\}$.
 3:     for each sliding window $SW_i$ in the set $S$ do
 4:         if $\mathrm{Drift\_Detection}(SW_i,\ SW_{i+1})$ == $\mathrm{True}$ then          ▷ Algorithm 2
 5:             Add $SW_{i+1}$ to the Adaptive Sliding Window $(\mathrm{ASW})$.
 6:         else
 7:             Predict on $SW_{i+1}$.
 8:         end if
 9:     end for
10: return $\mathrm{ASW}$
11: end procedure
12: procedure $\mathrm{Scenario\_Identification(Thresh\_DR,\ Thresh\_FAR)}$
13:     Perform Random Forest classifier on $SW_i$
14:     $(\mathrm{DR},\ \mathrm{FAR}) \leftarrow \mathrm{PredictWithCurrentModel}(SW_{i+1})$
15:     if $\mathrm{DR} < \mathrm{Thresh\_DR}$ then
16:         $\mathrm{scenario} \leftarrow 1$
17:     else if $\mathrm{FAR} > \mathrm{Thresh\_FAR}$ then
18:         $\mathrm{scenario} \leftarrow 2$
19:     end if
20: return $\mathrm{scenarios}$
21: end procedure
22: procedure $\mathrm{Handling\_Concept\_Drift(Testing\_Data,\ Thresh\_DR,\ Thresh\_FAR)}$
23:     $\mathrm{scenario} \leftarrow \mathrm{SCENARIO\_IDENTIFICATION(Thresh\_DR,\ Thresh\_FAR)}$
24:     $\mathrm{ASW} \leftarrow \mathrm{ADAPTIVE\_SLIDING\_WINDOW(Testing\_Data)}$
25:     $X \leftarrow \mathrm{PairwiseDistanceCalculation(ASW)}$
26:     $(c_x = \{c_1,\ c_2,\ c_3,\ c_4,\ c_5\}) \leftarrow \mathrm{KMeans}(X,\ k = 5)$
27:     $k_x \leftarrow \mathrm{Lowest\_Variance}(c_x)$
28:     $(\mathrm{normal},\ \mathrm{attacks}) \leftarrow \mathrm{PredictWithCurrentModel}(k_x)$
29:     if $\mathrm{scenario} == 1$ then
30:         $\mathrm{new\_attack\_instances} \leftarrow \mathrm{normal}$
31:         $\mathrm{current\_model} \leftarrow \mathrm{UpdateRandomForestModel(new\_attack\_instances)}$
32:     else if $\mathrm{scenario} == 2$ then
33:         $\mathrm{new\_normal\_instances} \leftarrow \mathrm{attacks}$
34:         $\mathrm{current\_model} \leftarrow \mathrm{UpdateRandomForestModel(new\_normal\_instances)}$
35:     end if
36: end procedure
```

Section 2 presents the background and motivation of this research and the concept drift phenomenon. Section 3 describes the proposed adaptive model and the concept drift detection method. Section 4 presents and discusses the experimental results in different settings. Section 5 outlines the conclusions and future work.

2 Background

In dynamic data environments, the challenge of concept drift becomes increasingly significant. As data distributions shift over time, predictive models can struggle to maintain their accuracy, as they are

Algorithm 2 Concept drift detection method

```
Input:   Sliding windows: $SW_i$, $SW_{i+1}$.
         Threshold      : $\mathrm{threshold}$.
Output:  Concept drift detection (Boolean).
 1: procedure $\mathrm{PCA}(SW_i,\ SW_{i+1})$
 2:     $pcs_i \leftarrow PCA(SW_i)$
 3:     $pcs_{i+1} \leftarrow PCA(SW_{i+1})$
 4:     $d_1 \leftarrow pcs_i[1]$
 5:     $d_2 \leftarrow pcs_i[2]$
 6:     $d_3 \leftarrow pcs_{i+1}[1]$
 7:     $d_4 \leftarrow pcs_{i+1}[2]$
 8: return $(d_1, d_2, d_3, d_4)$
 9: end procedure
10: procedure $\mathrm{Spearman\_Calculation}(pc1_i,\ pc2_i,\ pc1_{i+1},\ pc2_{i+1})$
11:     $\rho_1 \leftarrow \mathrm{SpearmanCorr}(d_1, d_2)$
12:     $\rho_2 \leftarrow \mathrm{SpearmanCorr}(d_3, d_4)$
13: return $(\rho_1,\ \rho_2)$
14: end procedure
15: procedure $\mathrm{Drift\_Detection}(SW_i,\ SW_{i+1},\ \mathrm{threshold})$
16:     $(d_1,\ d_2,\ d_3,\ d_4) \leftarrow \mathrm{PCA}(SW_i, SW_{i+1})$
17:     $(\rho_1,\ \rho_2) \leftarrow \mathrm{Spearman\_Calculation}(d_1,\ d_2,\ d_3,\ d_4)$
18:     if $(\delta \leftarrow (\rho_1 - \rho_2) < \mathrm{threshold})$ then
19:         Concept drift
20:     else
21:         Concept drift not detected
22:     end if
23: end procedure
```

often trained on static datasets that do not account for evolving patterns [19].

Concept drift presents substantial difficulties for traditional systems that rely on static models, making it essential to develop adaptive models that can adjust to these changes in real-time [16]. Addressing these challenges is crucial for ensuring the robustness and effectiveness of predictive models in the face of evolving data patterns [9]. There are two primary approaches to identifying concept drift within data streams:

1. Data distribution-based methods.
2. Error rate-based methods [9].

Data distribution-based methods monitor changes in data distribution over time by analyzing statistical properties [16]. For this, they rely on windowing methods, particularly fixed sliding windows, which serve as snapshots in time, denoted as t_0.

These snapshots facilitate the comparison of distributions P_{t_0} and P_{t_1}, enabling the detection of changes in data patterns, indicating concept drift [19]. Qahtan et al. [14] proposed employing Principal Component Analysis (PCA) alongside the Kullback-Leibler change score to detect abrupt changes and evaluate the severity of concept drift between sliding windows.

Wahab [2] proposed using PCA to detect and measure the degree of drift between two windows by evaluating the angle of intersection between their eigenvalues. If this angle is greater than or equal to 60°, a drift is signaled.

Chu et al. [5] introduced an ensemble approach to concept drift detection using

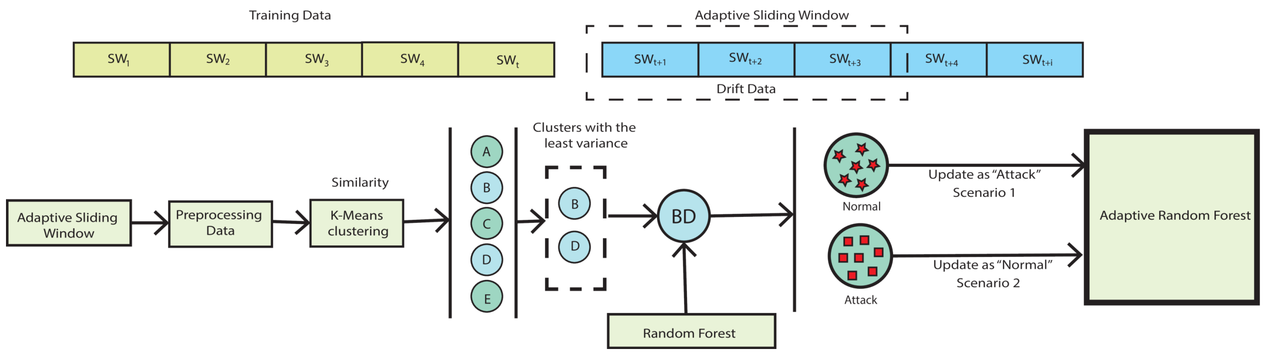


Fig. 2. Hybrid approach: K-means and RF for effective concept drift handling

nonparametric statistical metrics, including the Kolmogorov-Smirnov, Wilcoxon rank-sum, and Mann-Kendall tests, to compare the cumulative distribution functions of two windows.

Drift is detected by significant differences (p-values below a threshold) between two consecutive windows. However, despite their utility, fixed sliding windows present notable drawbacks: they may not effectively capture gradual changes in data distribution and can be inefficient due to their static nature [16, 19].

For instance, a window that is too large might incorporate outdated information, while a smaller window may lack sufficient context to accurately reflect changes. In contrast, adaptive windowing methods, such as Adaptive Windowing (ADWIN), dynamically adjust the size of the data window based on changes in data distribution.

ADWIN, for example, uses variance as its primary criterion for adjusting the window, allowing for a more flexible response to evolving data patterns [16]. Error rate-based methods monitor changes in predictive model performance by measuring accuracy or error rates over time [8].

Examples include Drift Detection Method (DDM) and Early Drift Detection Method (EDDM), which determine concept drift by tracking performance degradation [12]. Jain and Kaur [8] proposed monitoring both accuracy and False Alarm Rate (FAR), indicating drift when accuracy falls below a specified threshold and FAR exceeds another threshold.

Yang and Shami [19] proposed combining sliding and adaptive window techniques with error analysis. This method retains a significant amount of data related to concept drift in the sliding window when accuracy drops and uses an adaptive window for model retraining.

Error rate-based methods are particularly useful when labeled data is available, as they directly assess model performance on known outcomes [19]. After detecting drift, it's crucial to manage the changes effectively so the learning model can adapt seamlessly to new data patterns.

To address the concept drift, various strategies have been proposed to develop more robust IDS. Supervised adaptive approaches are often the most effective and quickest to adapt and detect attacks [3]. Seth, Singh, and Chahal [16] implemented an Adaptive Random Forest (ARF) classifier with ADWIN for IDS, demonstrating superior performance compared to Naive Bayes and K-Nearest Neighbors in a comparative analysis using streaming data. Chouchen and Jemili [4] proposed an ensemble adaptive approach combining ARF and Support Vector Regression (SVR) with ADWIN to enhance detection precision, employing a query strategy to guide the supervised module in identifying unknown attacks through expert intervention.

A Light Gradient Boosting Machine (LGBM) combined with an Optimized Adaptive Sliding Window (OASW) drift detector was proposed by Yang and Shami [19], achieving high accuracy while minimizing time and memory usage. Supervised adaptive approaches are effective but require labeled network traffic data, which is hard to obtain, especially with large data volumes.

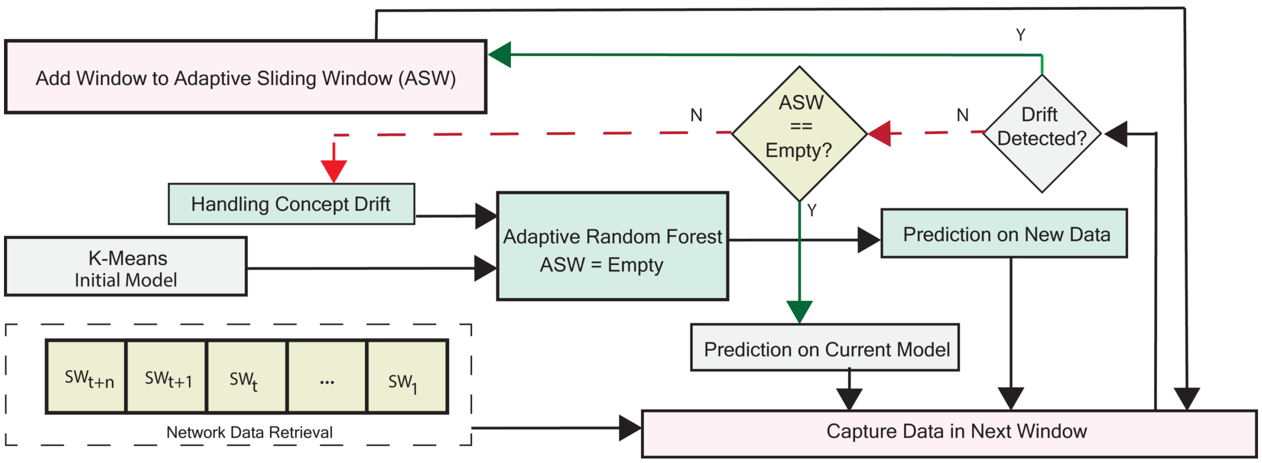


Fig. 3. Overview of the proposed adaptive model with concept drift detection

Also, the constant emergence of new network attack forms complicates maintaining an updated label database, hindering supervised models from effectively identifying and responding to emerging threats in network security [18].

This limitation highlights the need to explore unsupervised adaptive approaches, which are effective when labeled data is scarce or imperfect. Clustering techniques, which group instances based on similarities rather than labels, offer valuable insights and improve the detection of network attacks [3, 18]. However, clustering online poses challenges, including computational complexity in updating cluster centroids, sensitivity to parameter settings, and difficulties in handling concept drift [3]. Additionally, clustering methods may struggle with the high-dimensional and evolving nature of streaming data, potentially leading to performance degradation [15, 18].

Recent studies demonstrate that hybrid approaches, such as K-Means clustering combined with a Support Vector Machine (SVM) classifier [9] and K-Means clustering with Random Forest and Logistic Regression classifiers [8], effectively detect anomalies and concept drifts in real-time data streams.

These combinations leverage the strengths of each method, providing robust and accurate detection, flexibility, and efficient resource utilization by updating only when a concept drift is detected. Despite these advantages, reliance on unsupervised methods can lead to a high rate of false alarms, which may hinder the effective adaptation of IDS. Challenges such as data noise and class imbalance further negatively impact detection accuracy and overall system performance [8, 9].

2.1 Summary

Supervised adaptive approaches effectively detect and adapt to attacks, with the ARF classifier excelling in complex datasets [4, 16]. Hybrid approaches, which combine K-Means clustering with supervised learning, enhance performance by leveraging K-Means clustering's efficiency and its capability to operate without labeled data [8, 9]. This approach updates models only when concept drift is detected, which is more resource-efficient compared to continuous clustering methods [3, 18]. Nevertheless, since these approaches utilize unsupervised learning for model updates, they may experience reduced performance due to data imbalance and the presence of noisy data [8, 9].

Error rate-based concept drift detection typically requires ground-truth labels for optimal performance [19]. However, it can be adapted to unsupervised methods by using K-Means clustering for data labeling [8]. Combining this approach with distribution-based methods

Table 1. NSL-KDD dataset attack types

DoS	Probe	R2L	U2R
apache2, back, land, **mailbomb**, neptune, pod, **processtable**, smurf, teardrop, **udpstorm**	ipsweep, **mscan**, nmap, portsweep, **saint**, satan	spy, warezclient, ftp_write, guesspasswd, **httptunnel**, imap, multihop, **named**, phf, **sendmail**, **snmpgetattack**, warezmaster, **xlock**, **xsnopp**	bufferoverflow, loadmodule, perl, **ps**, rootkit, **snmpguess**, **sqlattack**, **worm**, **xterm**

and sliding window techniques enhances accuracy by effectively detecting data changes and concept drift.

This approach effectively detects both gradual and abrupt changes, even with limited labeled data [9].

Additionally, while distribution-based methods alone cannot quantify the severity of concept drift, they can be integrated with statistical metrics to achieve this [5, 14].

Finally, sliding windows enable continuous monitoring and adaptation, while adaptive windows retain extensive data patterns, thereby enhancing the model's performance and adaptability to new trends [16, 19].

3 Proposed Approach

The proposed approach for managing and detecting concept drift integrates two types of sliding windows to handle dynamic data: A fixed sliding window for stability in model performance and an adaptive sliding window for retraining to accommodate new data patterns. This adaptive mechanism enables the system to continually adapt and refine its understanding, ensuring robustness and accuracy in handling evolving data patterns.

The concept drift detection is founded upon the analysis of data distribution and error rate analysis, with a hybrid unsupervised approach being employed for managing the concept drift. Subsequent subsections detail the proposed adaptive model with concept drift detection.

3.1 Concept Drift Detection Method

The proposed method for concept drift detection harnesses PCA alongside Spearman's Correlation Coefficient (SCC). In data streams without drifts, the monotonic relationship is strong. The explanation is simple: during an attack, certain features such as connection time, energy consumption, and traffic level consistently rise on the targeted device.

However, in a firmware update, while values like connection time, energy consumption, and traffic levels may rise, irregular patterns do not necessarily follow suit. This alteration in the monotonic relationship, when evaluated between two sliding windows, may indicate the presence of concept drift and, furthermore, allow for the measurement of its severity.

SCC [1] is a statistical measure that evaluates the strength and direction of the relationship between two ordinal or quantitative variables. Unlike Pearson's correlation coefficient, which assesses the linear relationship between variables, Spearman evaluates the relationship based on the ranks of variable values.

This feature allows it to capture relationships that do not necessarily follow a linear pattern. The formula for SCC is given by Eq. 1:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}, \tag{1}$$

where ρ is the SCC, d_i are the differences between the ranks of the two variables, and n is the number of observations. PCA [7] is a statistical technique used to reduce the dimensionality of a dataset while preserving most of its variability. It

Table 2. Concept drift indicator between SW_0 and subsequent windows

SW_0	SW_1	SW_2	SW_3	SW_4	SW_5	SW_6	SW_7	SW_8	SW_9	SW_{10}
(δ)	0.0127	0.0584	0.0073	0.2550	0.2871	0.0274	0.0122	0.0059	0.2333	0.0284

achieves this by transforming the original variables into a new set of variables, called principal components, which are linear combinations of the original variables.

These principal components are ordered in terms of the amount of variability they explain in the data, with the first component explaining the most variability, followed by the second, and so on. In the proposed method, PCA is applied to each fixed sliding window SW_i to extract the two principal components PC_1 and PC_2.

These components represent the most significant variance in the data. SCC is then used to measure the monotonic relationship between these two components. For each sliding window SW_i, PCA is performed to reduce the dimensionality of the data to the two principal components PC_1 and PC_2. SCC (ρ_i) is computed between PC_1 and PC_2 within SW_i.

SCC evaluates the monotonic relationship, capturing both linear and non-linear correlations. In data streams without changes in the underlying concept, SCC is expected to remain constant across different sliding windows (SW_i, SW_{i+1}).

A significant change (δ) in SCC between (ρ_1) and (ρ_2) indicates potential concept drift. A large (δ) may suggest an abrupt change, whereas a small (δ) could represent a gradual change. Therefore, (δ) can serve as an indicator of the severity of the concept drift.

Algorithm 2 continuously monitors the data stream, applies PCA for dimensionality reduction (see lines 1-9 in Algoritm 2), and uses SCC to evaluate the monotonic relationship between the principal components (see lines 10-14 in Algorithm 2). A significant change in SCC between consecutive windows indicates concept drift (see lines 18-19 in Algoritm 2).

For practical implementation, selecting the sliding window size and the threshold (δ) for detecting significant changes in SCC is crucial. These parameters should be optimized based on the specific characteristics of the data stream and the application requirements.

3.2 Enhancing Drift Handling with K-means and Random Forest

To address concept drift, the adaptability of K-Means and Random Forest to evolving data patterns over time is leveraged. A comprehensive strategy for enhancing drift handling is outlined in Algorithm 1, providing a detailed process. Firstly, the data are divided into fixed sliding windows of size n, forming a set:

$$S = (SW_1, SW_2, SW_3, \cdots, SW_n). \quad (2)$$

Subsequently, Algorithm 2 is applied to precisely detect concept drift between two consecutive sliding windows $(SW_i,\ SW_{i+1})$. To effectively manage potential concept drift propagation to consecutive fixed sliding windows, an adaptive sliding window is utilized to retain these windows and extract the necessary information (see lines 1-11 in Algorithm 1).

The proposed approach suggests using the adaptive sliding window to identify clusters with similar statistical distributions. To achieve this, a pairwise similarity matrix is constructed using the Manhattan distance (see line 25 in Algorithm 1). This metric is selected for its strong generalization to higher dimensions, efficiency in parallel processing environments, and reduced computation time [13], all of which are crucial for the implementation of real-time adaptive IDS models. K-Means clustering is used to partition the pairwise distance matrix into k clusters, with k being a user-defined parameter (see line 26 in Algorithm 1). This strategy ensures that the clustering process not only organizes the data into meaningful clusters but also captures the inherent relationships between instances, leasing to more accurate and insightful results [6].

Table 3. Performance evaluation before drift handling (NSL-KDD dataset)

SW Id	Accuracy	Precision	DR	FAR	F1-score	Drift
(SW_0, SW_1)	0.9988	0.9985	0.9985	0.001	0.9985	No
(SW_1, SW_2)	0.9990	0.9995	0.9980	0.0003	0.9987	No
(SW_2, SW_3)	0.9990	1	0.9974	0	0.9987	No
(SW_3, SW_4)	**0.4986**	**0.9596**	**0.4021**	**0.0748**	**0.5667**	**Yes**
(SW_4, SW_5)	**0.4904**	**0.9623**	**0.3930**	**0.0695**	**0.5580**	**Yes**
(SW_5, SW_6)	0.9865	0.9953	0.9715	0.0031	0.9833	No
(SW_6, SW_7)	0.9832	0.9990	0.9608	0.00006	0.9795	No
(SW_7, SW_8)	0.9884	0.9979	0.9732	0.0013	0.9854	No
(SW_8, SW_9)	**0.5136**	**0.9609**	**0.4216**	**0.0764**	**0.5861**	**Yes**
(SW_9, SW_10)	0.9820	0.9924	0.9310	0.0050	0.9775	No
Average	**0.8439**	**0.9865**	**0.8047**	**0.0231**	**0.8632**	**—**

To determine the optimal value for k, Canopy clustering is employed. This method evaluates various potential values for k and suggests initial cluster centers [9]. The proposed model addresses two scenarios related to concept drift: (1) instances previously classified as normal are now attack, reducing the Detection Rate (DR) and indicating new threats; (2) instances previously attack are now normal, increasing the FAR and suggesting changes in network behavior.

To effectively define these scenarios, a concept drift detection method based on error rates, specifically focusing on DR and FAR, is proposed as outlined in Algorithm 1 (see lines 12-21 in Algorithm 1). Given the absence of ground-truth labels, clusters generated by the K-Means clustering are utilized to assign labels [8, 9].

This approach facilitates the effective determination of each scenario type, allowing for a robust identification and response to evolving data patterns.

If the DR metric decreases, it is crucial to identify attack instances that the current model fails to recognize. To address this, clusters with the least variance are selected to focus on significant examples according to their similarity, reducing sample size and managing data imbalance.

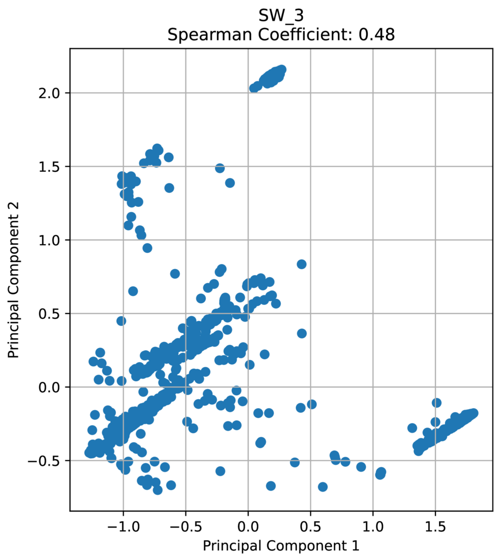
Previous research shows that clustering enhances model performance by focusing on representative examples, effectively addressing data imbalance [8, 9]. Analyzing these low-variance clusters helps differentiate between correctly and incorrectly classified distributions.

A Random Forest classifier is used to classify these low-variance clusters into normal and attack categories. When concept drift occurs, it reveals new attacks classified as normal, allowing for model refinement (see lines 29-31 in Algorithm 1).

When the FAR increases, previously classified attack instances are reassessed to improve normal instance classification (see lines 32-34 in Algorithm 1). Fig. 2 illustrates the model's handling of concept drift in both scenarios.

3.3 Adaptive Random Forest Training and Prediction

The primary aim of this study is to detect anomalies in network traffic. For this purpose, an ARF classifier was chosen due to its effectiveness in detecting and adapting to attacks [11, 16]. Due to the presence of concept drift, a model trained once cannot be reliably applied at all times, necessitating retraining upon detecting drift.



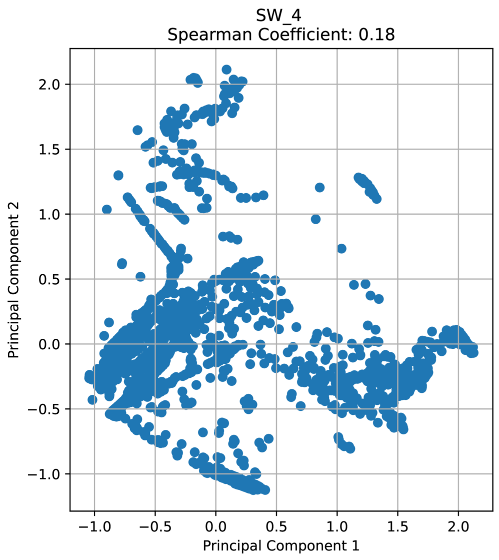


Fig. 4. Data distribution of principal components and SCC of SW_3 and SW_4

To ascertain drift in traffic, an SCC between the principal components of consecutive sliding windows is calculated over time. Fig. 3 provides an overview of the proposed approach for managing concept drift and detecting anomalies. The next section provides the results obtained using the proposed adaptive model.

4 Results and Discussions

In this research paper, an adaptive model for IDS is proposed, utilizing a hybrid approach based on k-Means clustering and Random Forest classification with concept drift detection. To validate the performance of the proposed adaptive model, several experiments were conducted using evaluation metrics such as accuracy, precision, DR, FAR, and F1-score [9].

The primary aim of the experimental framework is to evaluate the adaptive model's ability to adjust to evolving data and detect concept drifts. The experiments were performed using hardware resources, specifically an Intel® Xeon® CPU @ 2.20GHz and 12.67 GB of RAM. K-Means clustering was used to label the unlabeled dataset.

The resulting clusters served as input for a Random Forest classifier to develop the initial model, trained on the labeled training set from the NSL-KDD dataset. The NSL-KDD dataset [17] is the most widely preferred publicly available dataset for researchers in IDS [8, 9, 19].

It contains 148,517 records, each with 41 features plus a label class as the 42nd feature. Of these 41 features, 32 are continuous, while 9 are categorical. The NSL-KDD dataset is divided into training and test sets. The training set includes 22 attack types, categorized into four classes:

Denial of Service (DoS), User to Root (U2R), Remote to Local (R2L), and Probe. Additionally, 17 more attacks in the same classes are included in the test set. Table 1 shows the attack types in the test and training sets; bold text denotes the attacks introduced in the test set, while italic text denotes those exclusive to the training set.

Hence, a notable occurrence is the sudden drift observed from the training set to the test set within the NSL-KDD dataset [19]. This phenomenon arises from the absence of certain unknown attacks in the training set, leading to misclassification of such attacks in the test set as normal traffic.

The presence of categorical attributes required the use of label encoding to convert these attributes into numerical values suitable

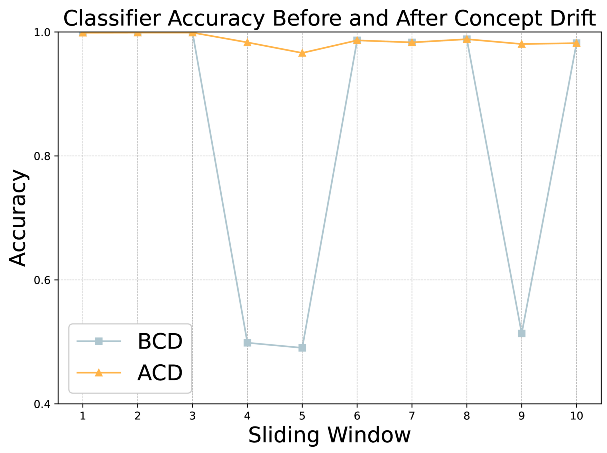


Fig. 5. Classifier accuracy before and after concept drift on NSL-KDD dataset

for analysis. Additionally, the range of all feature values was normalized using min-max normalization to ensure that all features contribute equally to the model's performance and improve convergence during training.

This preprocessing step is crucial for maintaining data integrity and enhancing the overall effectiveness of the proposed adaptive model. Furthermore, given the windowed nature of the data streaming concept, the integration of the two sets involved merging them into a unified dataset through the random selection of windows from each set.

For this study, a downsized NSL-KDD dataset comprising 5.0×10^3 instances was selected after testing various sliding window sizes to determine the optimal window size. It is important to note that larger windows might dilute concept drifts, while excessively small windows could compromise the effective detection of these drifts.

Therefore, selecting an appropriate window size is crucial for balancing detection sensitivity and maintaining model performance. After randomly merging the windows from the two sets, the objective is to detect concept drift between the sliding windows. This involves monitoring changes in the data distribution over time to identify shifts in patterns or relationships.

Table 2 presents the results obtained after employing the proposed concept drift detection method described in Subsection 3.1. As observed in Table 2, there are no significant changes (δ) in the SCC across sliding windows SW_0 through SW_3. However, this pattern is disrupted between windows SW_3 and SW_4, where a sudden change becomes noticeable.

This suggests a significant abrupt conceptual change, which persists in the subsequent window SW_5, and finally reappears in window SW_9. Additionally, as shown in Table 2, the value of (δ) in SW_9 is lower than in SW_4 and SW_5, which is reflected in the accuracy shown in Table 3. This indicates that the method is capable of detecting gradual changes.

Fig. 4 illustrates the data distribution of the two principal components within sliding windows SW_3 and SW_4, along with their SCC. In SW_3, a consistent pattern is evident, with points following a uniform trajectory, indicating a coherent relationship between the variables. However, in the presence of a concept drift in SW_4, this pattern diminishes, suggesting increased randomness or less predictability in the trajectory of the points.

This confirms the effectiveness of the proposed method for detecting concept drift. Tables 3 and 4 present the accuracy, precision, DR, FAR, and F1-score before and after drift handling in the NSL-KDD dataset. These results are then fed into the K-Means clustering, leading to the identification of five clusters based on the canopy clustering outcomes.

Table 3 highlights a significant decline in anomaly detection performance for (SW_3, SW_4), (SW_4, SW_5), and (SW_8, SW_9). This underscores the rationale for the proposed method in identifying the scenario based on the concept drift approach using error rates.

As observed, there is a decrease in the DR, highlighting its inability to recognize new attacks. Consequently, the prediction of the subsequent windowed data relied on the newly adapted model. Enhanced accuracy, precision, DR, FAR, and F1-score are evident in Table 4. These improvements observed further validate the method's capability in mitigating drift-induced performance degradation.

Furthermore, the columns Adaptation Time and Inference Time in Table 4 showcase the adaptation and inference times, respectively. It

Table 4. Performance evaluation after drift handling (NSL-KDD dataset)

SW Id	Accuracy	Precision	DR	FAR	F1-score	Adaptation Time	Inference Time
(SW_3, SW_4)	0.9832	0.9918	0.9874	0.0357	0.9896	25.04s	0.0590s
(SW_4, SW_5)	0.9662	0.9863	0.9721	0.0607	0.9792		
(SW_8, SW_9)	0.9806	0.9913	0.9848	0.0382	0.9880	14.41s	0.0863s

Table 5. Comparison of proposed model with existing approaches

Reference, Year	Method	Accuracy	Precision	DR	F1-Score	FAR	Approach
Bigdeli et al. [3], 2018	Incremental GMM	-	-	85%	-	7%	Unsupervised
Roshan et al. [15], 2018	ELM	-	-	77%	-	3.05%	Unsupervised
Yang and Shami [19], 2021	LGBM	98.31%	98.57%	98.30%	98.43%	-	Supervised
Jain and Kaur [8], 2021	RF, LR, and K-Means	93%	96%	94%	94%	9.60%	Hybrid U
Jain, Kaur, and Saxena [9], 2022	K-Means+SVM	91.33%	88.3%	91.7%	89.6%	2.11%	Hybrid U
Xiaolan et al. [18], 2022	EADNSD	91.80%	-	90.10%	-	7.60%	Unsupervised
Proposed	**K-Means+ARF**	**98.66%**	**99.52%**	**97.74%**	**99.78%**	**1.14%**	**Hybrid U**

is noteworthy that for (SW_3, SW_4) and (SW_4, SW_5), both adaptation and inference times exhibit identical values owing to the window retention by the adaptive sliding window. Experimental findings suggest that adaptation time for 1×10^4 instances exceeds prediction time for 5×10^3 instances, indicating the system's potential to process approximately 166.885 windows before real-time adaptation. Fig. 5 shows the evolution of accuracy over time. The graph shows accuracy fluctuations over time, with notable increases at specific intervals. For instance, a significant rise in accuracy occurs between the 4th and 5th data points, attributed to concept drift management. Table 5 provides a comparative analysis of the proposed adaptive model against various approaches found in the literature, utilizing the NSL-KDD dataset. The results demonstrate consistently superior performance across multiple metrics, underscoring the adaptive model's effectiveness and robustness in tackling intrusion detection challenges.

4.1 Discussion

Section 2 emphasizes the challenges faced by adaptive models for IDS, particularly in relying on supervised learning and labeled datasets for attack detection. The proposed adaptive model aims to mitigate this issue by demonstrating robust detection capabilities that are comparable to those of supervised methods, as demonstrated by the work of Yang and Shami [19].

Unlike unsupervised approaches, such as those proposed by Roshan et al. [15], Xiaolan et al. [18], and Bigdeli et al. [3], which continuously cluster real-time data, the proposed adaptive model identifies and manages concept drift only when necessary. This approach optimizes processing time, resource usage, and adaptability.

Hybrid approaches, such as those introduced by Jain, Kaur, and Saxena [9] and Jain and Kaur [8], integrate supervised and unsupervised methods to enhance detection performance while maintaining a low FAR.

In contrast, the proposed adaptive model focus on identifying patterns through clustering on the similarity matrix rather than directly analyzing raw data. This methodology facilitates the grouping of data distributions based on their similarities, allowing the detection of distributions familiar to the current model using Random Forest.

By applying a concept drift detection method based on DR and FAR, it becomes possible to identify distributions that the model fails to recognize, enabling timely model updates and enhancing overall performance. Additionally, employing an adaptive window helps retain critical information and manage the propagation of concept drift, enhancing pattern recognition and facilitating model updates.

The proposed concept drift method, based on distribution data, accurately identifies both abrupt and gradual changes by dynamically adjusting the threshold (δ). In contrast, traditional methods like ADWIN and PCA+Kullback-Leibler, as proposed by Qahtan et al. [14], are limited to detecting only abrupt changes and may miss subtler, gradual shifts.

5 Conclusion and Future Work

The concept drift detection method proposed in this research effectively identifies and quantifies the severity of concept drift, detecting both gradual and abrupt changes. This enhances the system's ability to recognize and adapt to evolving data patterns in real-time. Moreover, the adaptive IDS model described in this study effectively manages concept drift, leading to enhanced attack detection while maintaining a low FAR.

However, the experiments conducted indicate that a high adaptation time could potentially hinder the real-time implementation of these systems, representing a significant challenge. This is because it is crucial for the inference or prediction time to be very close to the adaptation time.

Future work will focus on optimizing the model at the algorithmic level and exploring hardware acceleration techniques to improve adaptation time while ensuring real-time detection capabilities.

Acknowledgments

The present work was supported by CONAHCYT/México (scolarship 986984). The authors thank the Universidad Iberoamericana Ciudad de México and the Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE).

References

1. **Abd-Al-Hameed, K. A. (2022).** Spearmans correlation coefficient in statistical analysis. International Journal of Nonlinear Analysis and Applications, Vol. 13, No. 1. DOI: 10.22075/ijnaa.2022.6079.

2. **Abdel-Wahab, O. (2022).** Intrusion detection in the IoT under data and concept drifts: Online deep learning approach. IEEE Internet of Things Journal, Vol. 9, No. 20, pp. 19706–19716. DOI: 10.1109/JIOT.2022.3167005.

3. **Bigdeli, E., Mohammadi, M., Raahemi, B., Matwin, S. (2018).** Incremental anomaly detection using two-layer cluster-based structure. Information Sciences, Vol. 429, pp. 315–331. DOI: 10.1016/j.ins.2017.11.023.

4. **Chouchen, I., Jemili, F. (2023).** Intrusion detection based on incremental learning. International Conference on Cyberworlds, pp. 448–455.

5. **Chu, R., Jin, P., Qiao, H., Feng, Q. (2023).** Intrusion detection in the IoT data streams using concept drift localization. AIMS Mathematics, Vol. 9, No. 1, pp. 1535–1561. DOI: 10.3934/math.2024076.

6. **Geng, X., Tang, H. (2020).** Clustering by connection center evolution. Pattern Recognition, Vol. 98, pp. 107063. DOI: 10.1016/j.patcog.2019.107063.

7. **Greenacre, M., Groenen, P. J. F., Hastie, T., D'Enza, A. I., Markos, A., Tuzhilina, E. (2022).** Principal component analysis. Nature Reviews Methods Primers, Vol. 2, No. 1. DOI: 10.1038/s43586-022-00184-w.

8. **Jain, M., Kaur, G. (2021).** Distributed anomaly detection using concept drift detection based hybrid ensemble techniques in streamed network data. Cluster Computing, Vol. 24, No. 3, pp. 2099–2114. DOI: 10.1007/s10586-021-03249-9.

9. **Jain, M., Kaur, G., Saxena, V. (2022).** A K-Means clustering and SVM based hybrid

ISSN 2007-9737

concept drift detection technique for network anomaly detection. Expert Systems with Applications, Vol. 193, pp. 116510. DOI: 10.1016/j.eswa.2022.116510.

10. **Kaspersky (2023).** Rising threats: Cybercriminals unleash 411,000 malicious files daily in 2023. www.kaspersky.com/about/press-releases/rising-threats-cybercriminals-unleash-411000-malicious-files-daily-in-2023.

11. **Leon, M., Markovic, T., Punnekkat, S. (2022).** Comparative evaluation of machine learning algorithms for network intrusion detection and attack classification. Proceedings of the International Joint Conference on Neural Networks, pp. 1–8. DOI: 10.1109/IJCNN55064.2022.9892293.

12. **Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., Zhang, G. (2019).** Learning under concept drift: A review. IEEE Transactions on Knowledge and Data Engineering, Vol. 31, No. 12, pp. 2346–2363. DOI: 10.1109/TKDE.2018.2876857.

13. **Pandit, S., Gupta, S. (2011).** A comparative study on distance measuring approaches for clustering. International Journal of Research in Computer Science, Vol. 2, No. 1, pp. 29–31. DOI: 10.7815/ijorcs.21.2011.011.

14. **Qahtan, A. A., Alharbi, B., Wang, S., Zhang, X. (2015).** A PCA-based change detection framework for multidimensional data streams: Change detection in multidimensional data streams. Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 935–944. DOI: 10.1145/2783258.2783359.

15. **Roshan, S., Miche, Y., Akusok, A., Lendasse, A. (2018).** Adaptive and online network intrusion detection system using clustering and extreme learning machines. Journal of the Franklin Institute, Vol. 355, No. 4, pp. 1752–1779. DOI: 10.1016/j.jfranklin.2017.06.006.

16. **Seth, S., Singh, G., Kaur-Chahal, K. (2021).** Drift-based approach for evolving data stream classification in intrusion detection system. Proceedings of the Workshop on Computer Networks and Communications, pp. 23–30.

17. **Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A. A. (2009).** A detailed analysis of the KDD CUP 99 data set. Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications, pp. 1–6. DOI: 10.1109/CISDA.2009.5356528.

18. **Xiaolan, W., Manjur-Ahmed, M., Nizam-Husen, M., Qian, Z., Belhaouari, S. B. (2022).** Evolving anomaly detection for network streaming data. Information Sciences, Vol. 608, pp. 757–777. DOI: 10.1016/j.ins.2022.06.064.

19. **Yang, L., Shami, A. (2021).** A lightweight concept drift detection and adaptation framework for IoT data streams. IEEE Internet of Things Magazine, Vol. 4, No. 2, pp. 96–101. DOI: 10.1109/IOTM.0001.2100012.

Article received on 29/05/2024; accepted on 04/07/2024.
**Corresponding author is Eric García-Huitzitl.*