Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.048105

ARTICLE

Adaptive Cloud Intrusion Detection System Based on Pruned Exact Linear Time Technique

Widad Elbakri[1], Maheyzah Md. Siraj[1,*], Bander Ali Saleh Al-rimy[1], Sultan Noman Qasem[2] and Tawfik Al-Hadhrami[3]

[1]Faculty of Computing, Universiti Teknologi Malaysia, Johor Bahru, Johor, 81310, Malaysia

[2]Computer Science Department, College of Computer and Information Sciences, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh, 11432, Saudi Arabia

[3]School of Science and Technology, Nottingham Trent University, Nottingham, NG11 8NS, UK

*Corresponding Author: Maheyzah Md. Siraj. Email: maheyzah@utm.my

Received: 28 November 2023 Accepted: 26 March 2024 Published: 20 June 2024

ABSTRACT

Cloud computing environments, characterized by dynamic scaling, distributed architectures, and complex workloads, are increasingly targeted by malicious actors. These threats encompass unauthorized access, data breaches, denial-of-service attacks, and evolving malware variants. Traditional security solutions often struggle with the dynamic nature of cloud environments, highlighting the need for robust Adaptive Cloud Intrusion Detection Systems (CIDS). Existing adaptive CIDS solutions, while offering improved detection capabilities, often face limitations such as reliance on approximations for change point detection, hindering their precision in identifying anomalies. This can lead to missed attacks or an abundance of false alarms, impacting overall security effectiveness. To address these challenges, we propose ACIDS (Adaptive Cloud Intrusion Detection System)-PELT. This novel Adaptive CIDS framework leverages the Pruned Exact Linear Time (PELT) algorithm and a Support Vector Machine (SVM) for enhanced accuracy and efficiency. ACIDS-PELT comprises four key components: (1) Feature Selection: Utilizing a hybrid harmony search algorithm and the symmetrical uncertainty filter (HSO-SU) to identify the most relevant features that effectively differentiate between normal and anomalous network traffic in the cloud environment. (2) Surveillance: Employing the PELT algorithm to detect change points within the network traffic data, enabling the identification of anomalies and potential security threats with improved precision compared to existing approaches. (3) Training Set: Labeled network traffic data forms the training set used to train the SVM classifier to distinguish between normal and anomalous behaviour patterns. (4) Testing Set: The testing set evaluates ACIDS-PELT's performance by measuring its accuracy, precision, and recall in detecting security threats within the cloud environment. We evaluate the performance of ACIDS-PELT using the NSL-KDD benchmark dataset. The results demonstrate that ACIDS-PELT outperforms existing cloud intrusion detection techniques in terms of accuracy, precision, and recall. This superiority stems from ACIDS-PELT's ability to overcome limitations associated with approximation and imprecision in change point detection while offering a more accurate and precise approach to detecting security threats in dynamic cloud environments.

KEYWORDS

Adaptive cloud IDS; harmony search; distributed denial of service (DDoS); PELT; machine learning; SVM; ISOT-CID; NSL-KDD

1 Introduction

Once an afterthought, cloud security has become paramount as this revolutionary IT paradigm, defined by NIST (National Institute of Standards and Technology) as accessible web-based computing resources, demands robust protection against evolving threats in a dynamically scalable environment [1]. Cloud computing delivers resources (SaaS, PaaS, IaaS) via private, public, hybrid, or community models, revolutionizing IT with on-demand scalability and flexibility [2]. Cloud's dynamism and flexibility boost efficiency but expose vulnerabilities to evolving attacks. Existing security solutions struggle to adapt, leaving businesses vulnerable to DDoS (Distributed Denial of Service), spoofing, and malware [3]. Evolving attacks overwhelm traditional defences, exploiting cloud environments' open, shared nature. Firewalls and authentication crumble as sophisticated threats constantly adapt [3]. Some DOS (Denial of Service) and DDoS attacks are intricate to detect using traditional methods [4]. Consequently, the development of more robust cloud security solutions becomes imperative.

Cloud computing's revolutionary adaptability comes at a cost: Increased vulnerability to ever-evolving cyberattacks. While traditional defences like firewalls and authentication crumble against these sophisticated threats, the open, shared nature of the cloud offers fertile ground for their proliferation. To fortify cloud security, a dedicated Cloud Intrusion Detection System (CIDS) is essential [5]. Acting as a vigilant watchdog, a CIDS monitors and identifies malicious activity through two main techniques: Signature-based and anomaly-based detection. Like a seasoned security guard, the former recognizes known attack patterns in incoming traffic, while the latter, an astute observer, identifies deviations from established normal network behaviour. Both approaches, working in tandem, form a robust shield against the ever-shifting landscape of cybercrime, ensuring the integrity and security of our cloud-powered world [6]. Signature-based detection excels at identifying known attacks with high accuracy and low false positives, but it falters against novel threats. Anomaly-based detection addresses this gap by creating a baseline of normal behaviour and flagging deviations, but it suffers from a higher rate of false alarms. Researcher Reacher demonstrated anomaly-based techniques' ability to detect known and unknown attacks across cloud tiers, but their tendency to generate false alarms remains challenging [7].

Cloud's constant flux, fueled by diverse users, scaling services, VMs, and auto-scaling, throws off traditional anomaly detection. The ever-shifting landscape of nodes constantly invalidates reference models, leaving security blind to new attack routes [8]. Cloud apps morph like chameleons under technical (VM scaling, upgrades) and non-technical (seasonality) influences. This volatility cripples traditional anomaly detection, leaving it chasing ghosts. Updating IDS models, especially during autoscaling's infrastructure upheavals, becomes as crucial as breathing-normalcy in the cloud is a moving target [9]. Cloud's chameleon-like behaviour, driven by technical (upgrades, migrations) and non-technical (seasonality, events) triggers, cripples static anomaly detection. Maintaining a constantly updated reference model becomes as vital as air–a necessity for effective intrusion detection in this ever-shifting landscape [10].

The cloud's superpower, its infinitely scalable muscle, lets it effortlessly adapt to user needs, keeping performance smooth even during traffic surges. This comes in two forms: Adding more virtual servers (horizontal scaling) or beefing up resources within existing ones (vertical scaling). However, this flexibility breeds a security paradox. The ever-shifting landscape throws off anomaly detection systems, leaving them struggling to tell friends from foes. Sudden resource spikes or seasonal traffic patterns can trigger false alarms, while cunning attacks weave through the confusion [11]. To tame this scalability beast and secure our cloud fortresses, we need an adaptive guard: A cloud-based Intrusion Detection System (IDS) that can pivot on a dime. We must strategically partition, distribute, and replicate data and constantly update reference models to reflect the cloud's ever-changing resource allocation and user behaviour. Only then can we dance with scalability without tripping into security disasters. It is a delicate ballet we must master to secure our cloud-powered future [12].

Several studies, including those by [9,13,14], have explored machine learning applications and the Negative Selection Algorithm (NSA) to enhance anomaly detection in cloud computing. Despite advancements, these approaches need more synchronized network tracking and effective IDS updates. Ibrahim and Zainal proposed an intrusion detection strategy utilizing a change point algorithm and machine learning for retraining [10]. However, the binary segmentation algorithm, which generates approximations and limits precise change point detection, could improve this method's effectiveness. Addressing adaptability issues during cloud scaling, this study introduces the Adaptive Cloud Intrusion Detection System (IDS) as an innovative solution.

While the cloud boasts near-limitless scalability, it throws a wrench in anomaly detection. The constant shuffle of virtual machines (VMs) makes establishing a stable "normal" profile nearly impossible, hindering effective threat identification. Scaling scenarios further complicate matters as the underlying infrastructure undergoes significant changes that demand frequent IDS model updates [15]. Existing adaptive CIDS solutions often stumble in this dynamic environment. Their reliance on approximations and imprecise change point detection methods leaves them vulnerable to false positives (legitimate scaling mistaken for attacks) [16].

Furthermore, traditional benchmark datasets like NSL-KDD must capture the intricacies of real-world cloud threats. This research fills these critical knowledge gaps by proposing ACIDS-PELT, a novel approach that integrates the Pruned Exact Linear Time algorithm and Support Vector Machine. Comprehensive evaluations across diverse datasets (ISOT-CID and DDoS) demonstrate ACIDS-PELT's superior performance in cloud intrusion detection. Its precise change point detection and reduced reliance on approximations are crucial to unlocking the full potential of secure cloud scalability.

The Key Contributions of this paper are:

- Combines algorithms for precise anomaly detection in cloud networks.
- Outperforms existing solutions thanks to detailed evaluations.
- Tackles key challenges of approximation and imprecision.
- Deepens understanding of cloud security complexities.
- Fills critical knowledge gaps through innovation and validation.

ACIDS-PELT unlocks the potential of secure cloud scalability.

The proposed solution comprises four key components: Feature selection utilizing Harmony Search Optimization and the Symmetrical Uncertainty (HSO-SU) Filter, a monitoring system leveraging the Pruned Exact Linear Time (PELT) change point detection algorithm, a Support Vector Machine-Based Adaptive Cloud Intrusion Detection System, and the training and testing modules. Particularly noteworthy is the precise change point detection technique employed by the PELT method, enabling meticulous updates to the IDS's reference model in response to profile fluctuations.

The paper adopts a clear and structured approach to presenting its findings. Section 2 delves into prior research, laying the groundwork for the proposed system. Section 3 then unveils ACIDS-PELT in detail, thoroughly examining the datasets and evaluation metrics used in Section 4. Section 5 presents the experimental results, while Section 6 compares ACIDS-PELT's performance against existing techniques. A comprehensive discussion analyses the results in Section 7, drawing comparisons to related works and fostering deeper insights. The paper concludes in Section 8 with conclusive remarks that encapsulate the essence of the study.

2 Related Works

Current Cloud Intrusion Detection Systems (CIDS) often need help in the face of dynamic cloud environments, hindered by static reference profiles and imprecise change detection, leading to false positives and missed threats. Scaling these systems adds complexity and further complicates accurate intrusion detection. ACIDS-PELT breaks through these limitations with a holistic and adaptable solution. It dynamically updates the reference profile using the (PELT) algorithm, ensuring precise adaptation to network changes even as the cloud scales. This precise change detection enhances accuracy and scalability by eliminating the need for frequent updates based on imprecise signals.

Additionally, ACIDS-PELT's hybrid feature selection reduces data volume, further streamlining the system for efficient operation in dynamic and large-scale cloud environments. The combined strengths of dynamic updates, precise change detection, and data reduction make ACIDS-PELT a powerful contender for superior, adaptable, and scalable security in the ever-evolving cloud landscape. This comprehensive system significantly improves accuracy and adaptability in safeguarding cloud environments by integrating hybrid feature selection and SVM-based detection, as shown in Table 1. The following summarizes existing research and its limitations, effectively addressed by the ACIDS-PELT system.

Table 1: Summary of literature review

Study	Method	Strengths	Limitations
[8]	Distributed IDS with behavior & knowledge-based detection	Novel combination, efficient resource utilization	Adaptability challenges not addressed, lack of specified monitoring technology
[17]	Neural network-based adaptive IDS	Distributed, resource-efficient	Performance validation lacking, no specified algorithmic technique for monitoring
[18]	LOF-based adaptive IDS	Adaptive anomaly detection, efficient data structure	Signature-based approach vulnerable to evasion, performance validation missing
[14]	Adaptive IDS with negative selection algorithm	Efficient update of detector set	Limited to anomaly detection, performance evaluation lacking
[19]	Neuro-fuzzy IDS	Adaptive learning for parameter optimization	Increased complexity, performance evaluation missing
[20]	Hypervisor-layer anomaly detection	Adapts to changing network environment	Limited scope to the hypervisor layer, performance evaluation lacking
[21]	Hybrid sampling & SDN for anomaly detection	Adapts traditional IDS to cloud environment	Performance evaluation missing, may not handle complex attacks

(Continued)

Table 1 (continued)

Study	Method	Strengths	Limitations
[22]	Stackelberg game for adaptive detection	Optimal resource allocation for attack detection	High complexity, performance evaluation missing
[23]	Edge computing-based DIDS with intelligent false alarm reduction	Efficient processing reduced response time	Limited scope to edge devices, performance evaluation missing
[24]	WLI-FCM & ANN hybrid IDS for cloud	High accuracy, low false alarm rate	Not tested in real-world cloud scenarios, limited dataset
[9]	Distributed ML-based IDS for cloud	Seamless integration with edge network	Scalability and adaptability challenges not fully discussed
[25]	Hybrid IDS for cloud (signature & anomaly)	Enhanced coverage and accuracy	Not adaptive to evolving attack patterns
[26]	SVM-based anomaly detection NIDS for cloud	Efficient feature selection, reduced dimensionality	Scalability and adaptability implications not explored
[27]	Statistical anomaly detection & network filtering IDS for cloud	Prompt DDoS attack detection	Reference profile update strategy not explicitly described
[28,29]	Self-adaptive genetic algorithm ANIDS for cloud	High precision, reduced false positives	Potential limitation of not updating profiles not addressed
[3]	Double deep-Q learning-based cloud IDS	Autonomous detection of emerging attacks	Potential challenges in complex cloud environments not addressed
[30]	Association rule mining for intrusion detection in cloud	Novel method, efficient detection of anomaly behaviors	Performance in complex real-world scenarios not explored
[31]	Adaptive anomaly detection framework with migration & reinforcement learning	Enhanced accuracy for unknown anomalies	Potential limitations in complex real-world scenarios not explored
[32]	Hybrid clustering & classification for anomaly detection	High performance on benchmark datasets	Not adaptive, does not address evolving attacks
[33]	Improved squirrel search algorithm & Modified-deep belief network IDS for cloud	Efficient feature selection, binary and multi-class anomaly detection	Potential limitations in dynamic real-world scenarios not addressed
[34]	Hybrid IDS for cloud (signature & anomaly)	High performance on various datasets	Scalability and adaptability challenges not addressed

(Continued)

Table 1 (continued)

Study	Method	Strengths	Limitations
[35]	Backpropagation neural network for anomaly detection	Effective prediction of normal and abnormal behaviors	Potential limitations in complex real-world scenarios not explored
[13]	Adaptive ensemble random fuzzy (AERF) algorithm for cloud anomaly detection	Handles abnormal sample distributions	Update strategy for reference profile not addressed

Existing research tackles cloud intrusion detection from various angles:

- Distributed systems based on machine learning (ML) [8,17] offer scalability but need more specifics on monitoring cloud network changes or adaptation techniques.
- Adaptive cloud IDS with signature-based and anomaly-based detection [18,19] are vulnerable to evolving attacks and lack performance validation.
- Hypervisor-based anomaly detection [20,21] requires specific hardware and might not generalize well.
- Adaptive detection strategies using game theory and edge computing [22,23] are promising but complex and require further exploration.

Other approaches utilize:

- Non-adaptive hybrid clustering and classification [24] performs well on fixed datasets but struggles with evolving threats.
- Hybrid intrusion detection models integrating signature and anomaly techniques [25] demonstrate promising results but need further evaluation on scalability and adaptability.
- Hybrid algorithms combining ML and neural networks [26] show good accuracy but need more discussion on scalability and adaptability.

Limitations identified in existing research:

- Lack of focus on adaptability to dynamic cloud environments.
- Reliance on static reference profiles or insufficient update strategies.
- There needs to be more discussions on scalability and real-world applicability.

Our proposed ACIDS-PELT system addresses these limitations by:

- Monitoring and dynamically updating the reference profile.
- Employing precise change point detection for accurate updates.
- Integrating multiple techniques for robust anomaly detection.

3 The Proposed Adaptive Cloud Intrusion Detection System Based on PELT and SVM (ACIDS _PELT _SVM)

3.1 The Pruned Exact Linear Time (PELT) Algorithm

The ever-increasing reliance on cloud computing has led to a surge in security concerns due to the dynamic and distributed nature of cloud environments. CIDS plays a vital role in safeguarding cloud infrastructures from potential threats and attacks. However, the continuous evolution of attack patterns and changes in network topology present significant challenges for traditional IDS solutions. To address this issue, this paper proposes an innovative Adaptive Cloud Intrusion Detection System based on the (PELT) algorithm and Support Vector Machine (SVM)-ACIDS_PELT_SVM. The ACIDS_PELT_SVM is designed to dynamically adapt to the evolving cloud environment, detecting anomalies efficiently and accurately, thus providing enhanced security for cloud-based systems. In this paper, we present the architecture, working principles, and evaluation of ACIDS_PELT_SVM, comparing its performance with existing techniques and demonstrating its effectiveness in cloud intrusion detection.

The adaptive cloud intrusion detection system, depicted in Fig. 1, consists of four key components: The feature selection component employing hybrid Harmony Search Optimization and the Symmetrical Uncertainty Filter (HSO-SU) for relevant feature selection, the surveillance component using Pruned Exact Linear Point (PELT) to monitor network traffic data alterations and update the model as needed, and the training and testing components utilizing respective data for training and evaluation. Algorithm 1 illustrates the proposed Cloud IDS, and the procedure is presented in Algorithm 1. The (PELT) algorithm, a cornerstone of our surveillance component, is a powerful tool extensively used in various domains, including anomaly detection, signal processing, and time series analysis. Its ability to detect change points in data makes it particularly well-suited for monitoring alterations in network traffic patterns, enhancing the adaptability and responsiveness of our intrusion detection system. The subsequent section will delve into the details of the system's components and Algorithm 1.

3.1.1 Overview and Functionality

In data analysis, detecting changepoints in a sequence is a crucial task. Changepoints represent points in the sequence where the underlying data distribution undergoes a significant change. The (PELT) algorithm is a powerful method that efficiently identifies these changepoints. In this blog post, we will delve into the details of the PELT algorithm and understand how it is used to detect change points accurately.

3.1.2 Algorithm Outline

The PELT algorithm operates on a sequence (y_1) and requires a few essential components to work effectively. These components include the penalty parameter (penalty), the cost function (cost_function), and the minimum segment length (min_segment_length). Detail of exploring the algorithm's steps in the following section:

a) Initialization:

The algorithm begins by creating an empty list of changepoints. This list will be populated as the algorithm progresses, ultimately containing the identified changepoints in the sequence.

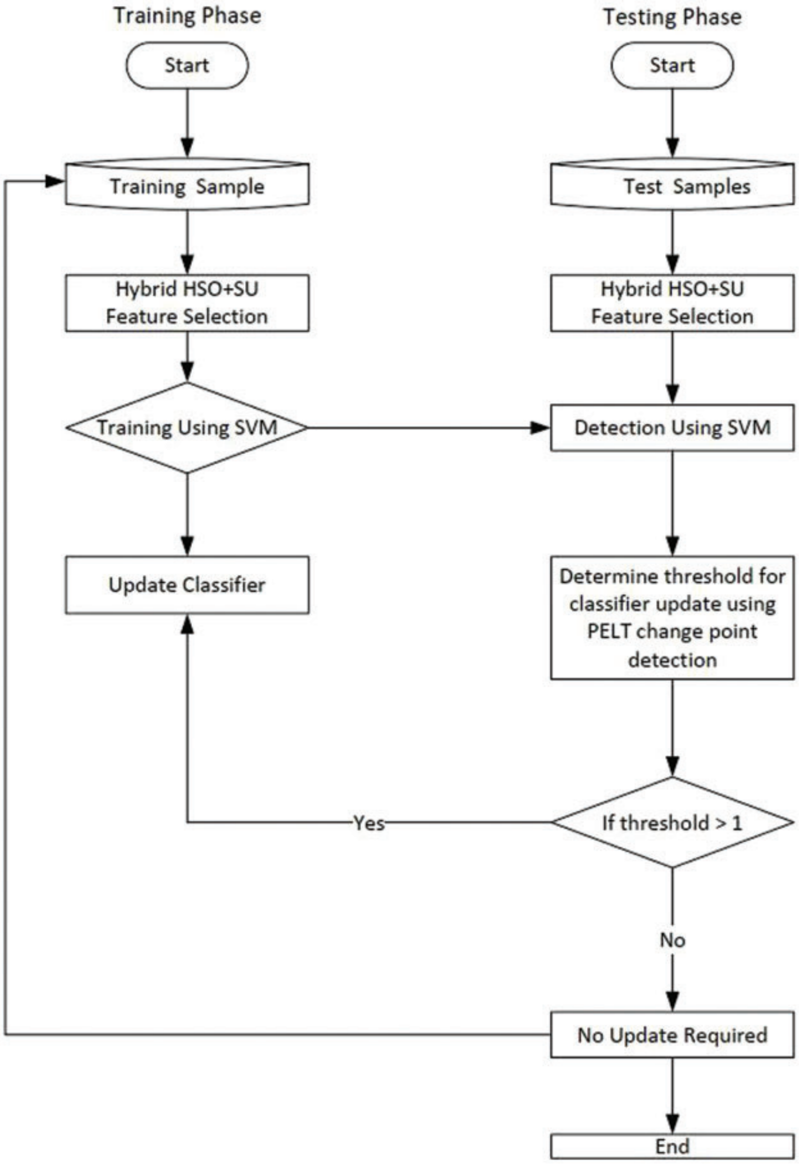


Figure 1: Activity diagram illustrating the functionality within the proposed CIDS-PELT framework

b) Define Cost Function:

The cost function (cost_function) is vital in the PELT algorithm. It quantifies the cost of each segment in the sequence, helping to determine the optimal locations for potential changepoints. The choice of the cost function depends on the specific characteristics of the data being analyzed.

c) Compute Pruned Exact Linear Time (PELT):

The main computation in the PELT algorithm involves calculating a series of partial sums, which are then pruned based on a pruned cost condition. This pruning process significantly reduces the computational complexity, making the algorithm highly efficient.

d) Identify Changepoints:

The PELT algorithm identifies potential changepoints in the sequence using the computed partial sums and the pruned cost condition. These potential changepoints are then evaluated based on the penalty parameter (penalty) and the minimum segment length (min_segment_length) to determine the final changepoint set.

e) Output:

The PELT algorithm produces a list of changepoints, indicating the locations in the sequence where significant changes occur. These changepoints can serve as valuable insights, helping analysts understand and interpret the underlying patterns in the data.

The PELT algorithm is a highly effective and efficient approach for detecting sequence changepoints [36]. Its capability to leverage penalty parameters, cost functions, and minimum segment lengths allows it to accurately pinpoint significant changes in the data. Due to its ability to handle large datasets and its time complexity of O(n), PELT is widely favored for applications such as anomaly detection, signal processing, and time series analysis [36]. Integrating the PELT algorithm into the data analysis pipeline offers valuable insights into the data's dynamics and trends, helping analysts understand and interpret the underlying patterns in the data.

Algorithm 1: For Pruned Exact Linear Time (PELT)

```
Input:
                A sequence of time series dataset $y_1, y_2, y_3 \ldots, y_n$ where $y_i \in R$.
                A cost function $C$.
               A penalty constant $\beta$ that prevent oervfitting.
               A constant $K$.
Output:
              A set of change point recorded cp.
Begin
Initialize: $cp = \varnothing$ and $S = \{[1, n]\}$, where n the length of the data.
Iterate for $\tau^\wedge * = 1, \ldots, n$:
  While $S \neq \varnothing$ do
     Select an element from S and denote it as $[s, t]$.
    Calculate $F(\tau *) = min_0 \leq \tau < \tau * [F(\tau) + C(y(\tau + 1): \tau *) + \beta]$.
    Set $\tau^\wedge = arg\{min_0 \leq \tau < \tau * [F(\tau) + C(y(\tau + 1): \tau *) + \beta]\}$.
   Set $cp(\tau *) = (cp(\tau^\wedge), \tau^\wedge)$.
   Set $R\_(\tau * +1) = \{\tau \in R\_(\tau *) \cup \{\tau *\}: F(\tau) + C(y\_(\tau + 1: \tau *)) + K \leq F(\tau *)\}$.
  Estimate change point position p of $[s, t]$.
       If $\lambda < C$, then no change is detected, remove $[s, t]$ from S.
         Else if $\lambda > C$, add p to cp.
       End if.
     End while.
    Return the set of change points cp.
   End.
```

3.2 System Components

In addressing the intricate challenges posed by dynamic and distributed cloud environments, our proposed solution, the Adaptive Cloud Intrusion Detection System (ACIDS-PELT), stands as a robust and innovative approach. ACIDS-PELT leverages a unique combination of the (PELT) algorithm and the Support Vector Machine to enhance intrusion detection in cloud systems. This adaptive model is specifically designed to overcome scalability hurdles, rapidly changing attack patterns, and limitations observed in existing solutions. Four essential components which are the following.

3.2.1 Feature Selection Component

Feature selection, an essential step in optimizing machine learning algorithms, entails removing irrelevant attributes to improve the accuracy and overall efficiency [5] of the algorithm's efficiency. Feature selection is required because of the inherent noise and insignificant data features in machine learning tasks [37]. The feature selection component in this study addresses the issue of noisy data even further, ensuring the robustness of the selected features for the proposed Adaptive Cloud Intrusion Detection System (ACIDS-PELT).

The hybrid harmony search algorithm (HSO) plays a pivotal role in the proposed feature selection process, leveraging its effectiveness in solving complex optimization problems. Developed as an innovative approach, HSO combines the strengths of harmony search algorithms with other optimization techniques, creating a synergistic method for tackling feature selection challenges. HSO excels in efficiently exploring and exploiting the search space, making it an attractive choice for selecting features in diverse datasets. The algorithm commences the feature selection task by representing the problem as a musical harmony improvisation, where each solution corresponds to a musician's note. The iterative steps involve harmonizing the best solutions to create improved melodies, simulating a musical ensemble, and seeking optimal compositions. This unique approach allows HSO to navigate the high-dimensional feature space, identifying the most relevant features for enhanced model performance [38].

These steps encompass the essence of the HS algorithm's operation, making it a compelling choice for feature selection in this study:

Step 1: Initialization: The population of candidate solutions representing feature subsets is initialized.

Step 2: Harmony Memory Consideration: A memory stores the best solutions, guiding the search towards favorable feature combinations.

Step 3: Harmony Construction: New harmonies are constructed by blending existing solutions with random adjustments, fostering diversity.

Step 4: Evaluation and Update: The fitness of each harmony is evaluated, and the memory is updated with superior solutions to preserve high-quality features.

Step 5: Termination Criteria: The search iterates until a predefined criterion is met, ensuring convergence to an optimal or near-optimal feature subset.

Our previous work [39] provides detailed insights into the harmony search attribute selection process, showcasing its effectiveness in identifying relevant features for the proposed Adaptive Cloud Intrusion Detection System (ACIDS-PELT). By integrating the Hybrid Harmony Search algorithm and the symmetrical uncertainty filter, our feature selection approach aims to optimize the system's performance, ensuring that the selected attributes contribute significantly to anomaly detection while mitigating the impact of noise and irrelevant data features. This hybridization strategy is a critical component of our comprehensive evaluation, aiming to demonstrate the efficacy of the chosen algorithm in addressing the unique challenges posed by cloud intrusion detection. ACIDS-PELT employs a Hybrid Harmony Search algorithm to meticulously select relevant features, distinguishing between normal and abnormal network traffic within the dynamic cloud environment. The following is the role and the impact of the component:

Role: This component plays a crucial role in enhancing the efficiency of anomaly detection. It identifies relevant features that effectively distinguish between normal and abnormal network traffic in a cloud environment.

Impact: Efficient feature selection reduces the dimensionality of the dataset, thereby enhancing the overall performance of the intrusion detection system (IDS). The hybrid harmony search algorithm ensures a comprehensive feature space exploration, contributing to anomaly detection accuracy.

3.2.2 Surveillance Component: Enhancing Security through Changepoint Analysis

In data analysis, changepoint detection is critical, especially in the context of cloud intrusion detection systems. It locates points or intervals in network traffic data where statistical properties significantly change [40]. Because cloud environments are dynamic and ever-changing, monitoring these changes is critical to identify anomalies and potential security threats. Changepoint analysis, made possible by algorithms such as the (PELT) in the Adaptive Cloud Intrusion Detection System (ACIDS-PELT), provides a precise mechanism for detecting shifts in the regular profile of network traffic data. This capability is critical for the system's adaptation to the changing cloud landscape, as it ensures timely updates to the reference model and accuracy in the face of dynamic changes. In a broader sense, accurate changepoint detection is a critical step in gaining insights into data dynamics and uncovering meaningful patterns or anomalies, significantly contributing to the effectiveness of intrusion detection mechanisms in cloud environments. The Importance of Monitoring Changes in the Cloud Environment.

It is critical to emphasize the significance of surveillance and changepoint analysis in a cloud intrusion detection system. The dynamic nature of cloud environments, as demonstrated by constant changes in user profiles, resource allocations, and virtual machine migration, emphasizes the importance of continuous monitoring. Surveillance, primarily through changepoint analysis, is becoming increasingly important in fortifying the Cloud Intrusion Detection Systems (CIDS) security framework. Changepoint analysis helps identify points or intervals where the statistical properties of network traffic data significantly change. In the complex dynamics of cloud environments, these changes may indicate anomalies or potential security threats. Using changepoint detection algorithms such as the (PELT) within the Adaptive Cloud Intrusion Detection System (ACIDS-PELT), the system can accurately identify shifts in the regular profile of network traffic data. This capability ensures the system's adaptability to the evolving cloud landscape, providing a robust defense against emerging security challenges. Therefore, surveillance and changepoint analysis emerge as indispensable components in ACIDS-PELT, addressing the complexities of cloud environments and bolstering the effectiveness of intrusion detection mechanisms.

The surveillance component of our proposed Adaptive Cloud Intrusion Detection System (ACIDS-PELT) leverages the (PELT) algorithm for changepoint detection. This crucial mechanism empowers the system to identify anomalies and potential security threats by meticulously monitoring fluctuations in network traffic data. Cloud environments, marked by continual changes like the movement of virtual machines (VMs), shifts in network traffic patterns, and system configurations, demand a vigilant and adaptive intrusion detection approach. Surveillance, primarily through changepoint analysis, is pivotal in fortifying the Cloud Intrusion Detection Systems (CIDS) security framework. Detecting changepoints in data analysis is fundamental for identifying shifts or abrupt changes in a dataset's underlying structure or characteristics. These points mark instances where the statistical properties of the data undergo significant alterations, such as mean, variance, or distribution. Understanding and pinpointing changepoints are essential for various applications, including anomaly detection, quality control, and time series analysis. In anomaly detection, changepoints help identify deviations from the norm, signaling potential irregularities or events of interest. Quality control relies on changepoint analysis to detect variations in manufacturing processes or product quality. In time series analysis, detecting changepoints aids in identifying shifts in trends, seasonality, or other patterns over time. Through the PELT algorithm, ACIDS-PELT offers a precise mechanism for detecting shifts in the regular profile of network traffic data, ensuring the system's adaptation to the evolving cloud landscape. ACIDS-PELT dynamically updates its reference model by systematically identifying changepoints and maintaining accuracy and effectiveness in dynamic alterations. In essence, surveillance and changepoint analysis are indispensable components of the ACIDS-PELT framework, providing a robust solution to the challenges posed by the intricate dynamics of cloud environments. In the following, we delve into the role and impact of each component:

Role: The surveillance component utilizes the (PELT) algorithm for changepoint detection in network traffic data, monitoring fluctuations in the regular profile to ensure the IDS's reference model is updated meticulously in response to dynamic shifts.

Impact: PELT's precise changepoint detection is fundamental to identifying anomalies and potential security threats. It contributes to the adaptability of the IDS, ensuring effective responses to changes in the cloud environment, including normal variations and emerging attack patterns.

The behaviour of cloud networks undergoes frequent modifications due to diverse user profiles, dynamic resource allocation, and the intricate process of VM migration. These changes impact the effectiveness of security monitoring mechanisms in identifying potential attacks. Consequently, cloud intrusion detection systems (IDS) must be tailored to navigate the complexities of cloud environments. In this context, an adaptive IDS emerges as a strategic solution, employing a dynamic approach to update either the standard reference model or the attack signature. The adaptability of an IDS can be rooted in anomaly-based or attack signature-based strategies. Anomaly-based adaptive IDS undergoes periodic retraining with new data to detect deviations from the regular profile, whereas signature-based IDS introduces new rules to the existing rule set. In the cloud infrastructure, the dynamic addition and removal of virtual networks and monitored nodes, each with unique security specifications, further accentuates the need for a flexible and responsive cloud-based IDS [41].

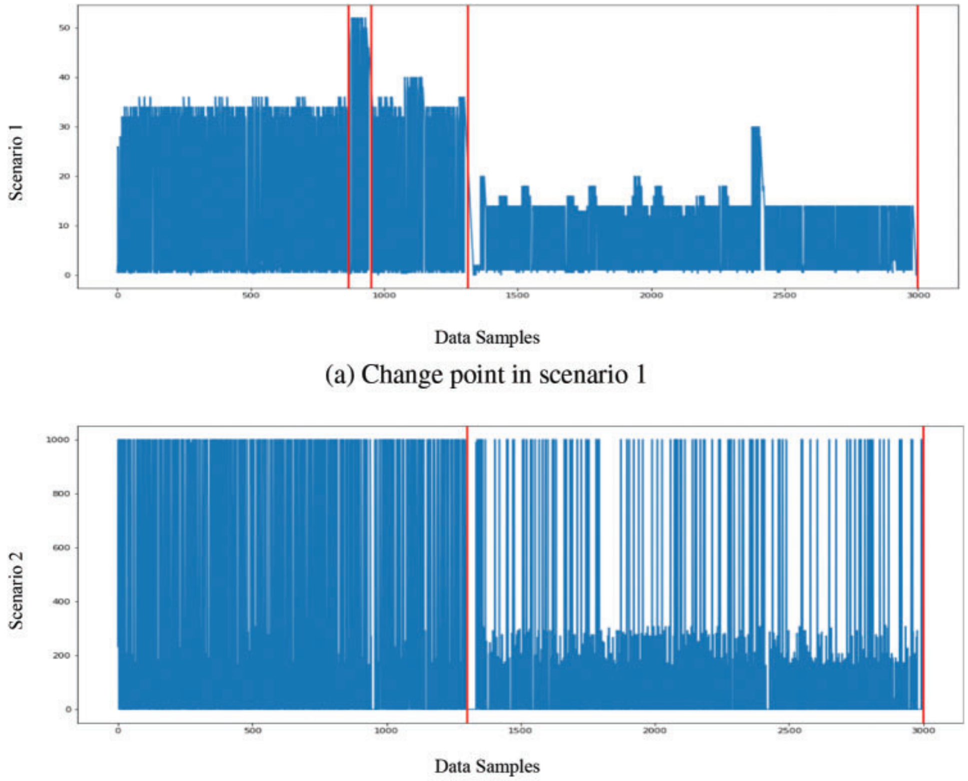
To address this demand for adaptability, Krishnan and Chatterjee proposed an anomaly-based adaptive IDS that emphasizes a pivotal monitoring feature capable of detecting alterations in the regular profile and adjusting it accordingly [8]. In line with this philosophy, the surveillance component of the Proposed Adaptive Cloud Intrusion Detection System (ACIDS-PELT) integrates the (PELT) change point detection algorithm. The essence of changepoint analysis revolves around identifying instances in a dataset where statistically significant changes occur. Formally, given an ordered data sequence $y_{1:n} = (y_1, \ldots, y_n)$ changepoint detection involves determining multiple transition points, denoted as m, and their respective locations $T_{1:m} = (T_1, \ldots, T_m)$. A common approach to defining these transition points involves minimizing Eq. (1) in the context of changepoint analysis.

$$ML\left(T_{1:m}\right) = \sum\nolimits_{i=1}^{m+1}\left[C\left(y_{\left(T_{i-1}+1\right):T_1}\right)\right] + Bf\left(m\right) \tag{1}$$

In the context of changepoint analysis, the symbol C denotes the cost function assigned to a particular segment, while $\beta f(m)$ serves as a regularization parameter aimed at mitigating the risk of overfitting. Among the array of available cost functions, the negative log-likelihood stands as the predominantly employed option, complemented by the utilization of Akaike's Information Criterion (AIC) and Bayesian Information Criterion (BIC) as commonly applied penalties [36].

The (PELT) algorithm emerges as an optimal partitioning technique within this framework. PELT methodically identifies subsequent changepoints based on the location of the previous change point. It achieves this by calculating the optimal value of the cost function for the optimal partition of the data preceding the last changepoint, coupled with the cost associated with the segment spanning from the last changepoint to the conclusion of the dataset. This methodology enables PELT to effectively determine significant changes while ensuring an optimal balance between the quality of fit and the complexity of the model.

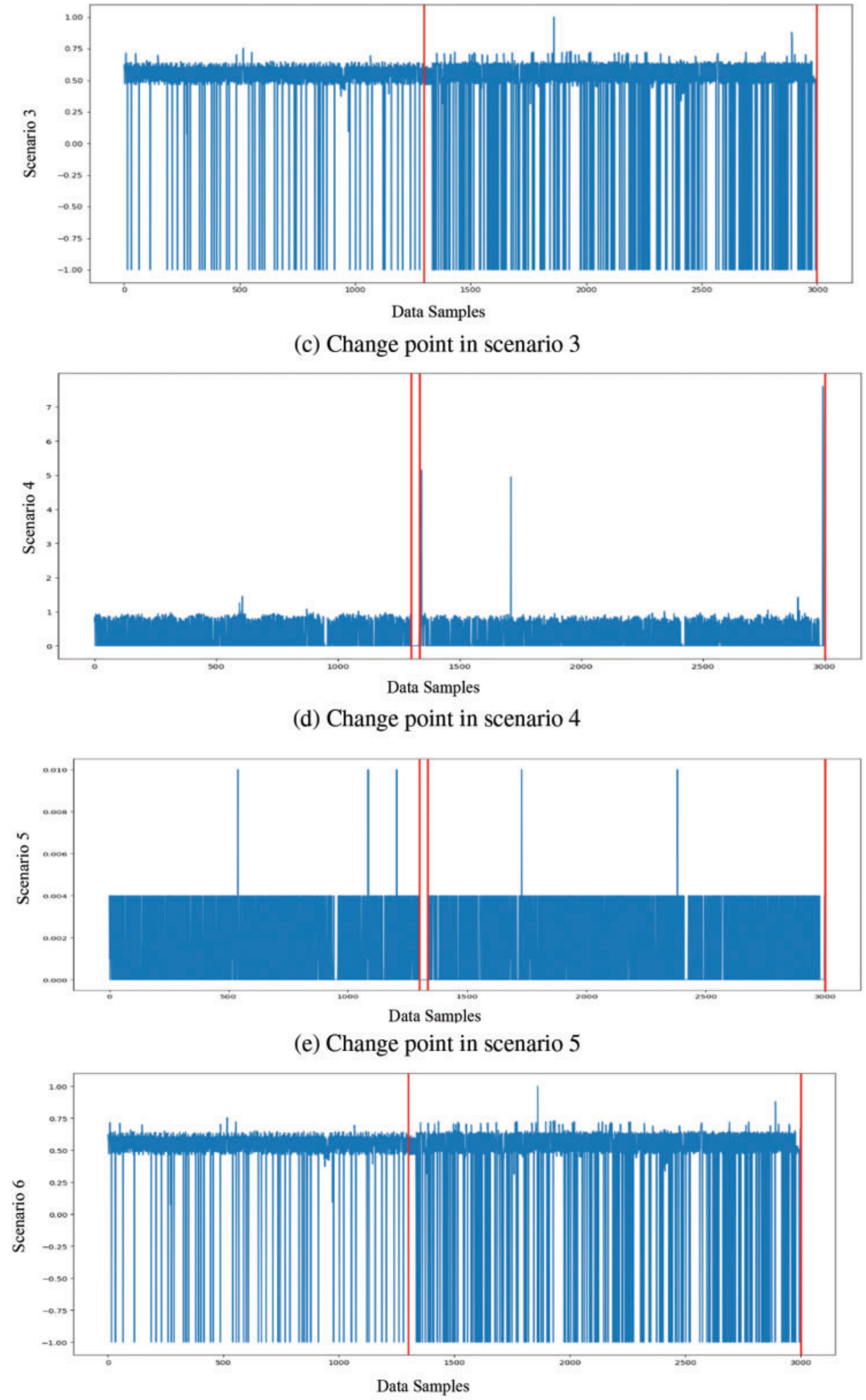
The surveillance component plays a pivotal role in establishing the criteria for model updates. To define the threshold for triggering updates, a careful analysis of the data was conducted over a specific observation period. In this context, a dataset comprising 4000 samples was gathered, and the mean change point for these samples was meticulously monitored, as visually depicted in Figs. 2 and 3. Notably, the graphical representation reveals that the peak change point predominantly falls within the range of 0 to 2. As a result, the decision was made to set the threshold for model updates at the average of these values (i.e., 1).



(a) Change point in scenario 1

(b) Change point in scenario 2

Figure 2: (Continued)



(c) Change point in scenario 3

(d) Change point in scenario 4

(e) Change point in scenario 5

(f) Change point in scenario 6

Figure 2: Detection of change points in scenarios 1 to 6 using the ISOT-CID dataset

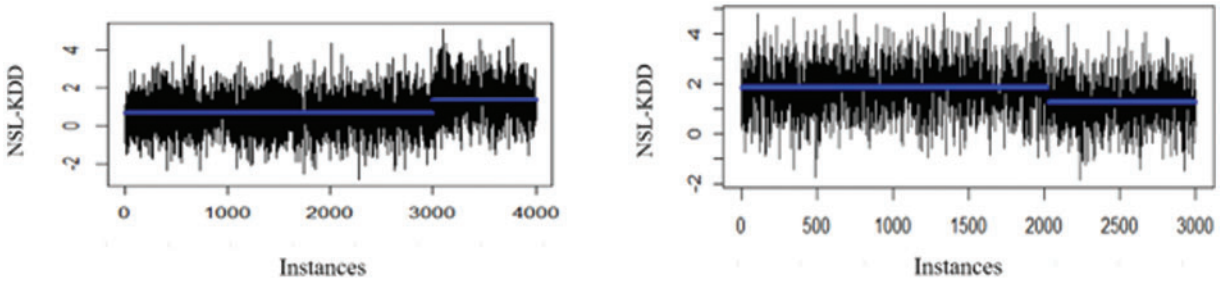


Figure 3: Detection of change points on sceneries 1 and 2 using the NSL-KDD dataset

Upon closer examination of Fig. 2, it becomes evident that the mean change point surpasses the designated threshold at both the 3000th and 4000th data samples. Furthermore, Fig. 3 illustrates a similar pattern where the mean change point once again surpasses the established threshold. Consequently, the classification model is configured to undergo an update each time the predefined threshold for updates is met, ensuring that the model remains responsive to evolving data dynamics.

3.2.3 Training Set Component

ACIDS-PELT relies on a labeled training set, comprised of network traffic data. This set is pivotal for training the Support Vector Machine classifier, allowing the system to learn and adapt to diverse patterns of normal and malicious activities. The role and the impact of the component are as follows:

Role: The training set consists of labelled network traffic data and is used to train the Support Vector Machine (SVM) classifier. The SVM must learn and establish patterns distinguishing between normal and malicious activities.

Impact: A well-trained SVM is essential for the accuracy of the intrusion detection system. The training set allows the model to generalize from labelled data, improving its ability to identify and classify security threats during testing.

The training and testing component of the system incorporates the Support Vector Machine (SVM) as its classification algorithm. SVM is a statistical learning method widely utilized for regression and pattern recognition tasks. Positioned within the realm of generalized linear classifiers, SVM was developed by Vapnik [42] and has garnered significant prominence within the machine learning domain. This acclaim is attributed to SVM's remarkable robustness in handling data characterized by noise and sparsity, rendering it a preferred choice across diverse machine-learning applications.

Support Vector Machine operates by projecting the input vector into a feature space, subsequently establishing a hyperplane with the maximal margin that effectively segregates positive and negative instances. In essence, SVM capitalizes on the margin maximization principle to facilitate classification. Notably, the versatility of SVM extends to nonlinear classification tasks as well, wherein input vectors may not be linearly separable. This is encapsulated by a general nonlinear SVM, as formulated in Eq. (2).

$$u = \sum_{j=1}^{N} y_j \alpha_j K\left(\vec{x}_{j,} \vec{x}\right) - b \tag{2}$$

In the context of Eq. (2), u represents the output of the Support Vector Machine (SVM). K is a kernel function corresponding to the kernel, which quantifies the resemblance between a given stored training example $\vec{x}_i$ to the input $\vec{x}$. The value assumes either $y_i \in (-1, +1)$, signifying the desired output of the classifier. The threshold is represented by the parameter b, while α_i are symbolizes the weights that harmonize the various kernels. In the case of linear SVMs, the kernel function K assumes a linear form. Consequently, Eq. (2) can be succinctly represented as:

$$u = \vec{w}.\vec{x} - b \tag{3}$$

where $\vec{w} = \sum i\, y_i\, \alpha_i\, \vec{x}_i$.

Training an SVM entails finding the α_i, and usually expressed as the minimization of a dual quadratic form:

$$\min_{\vec{\alpha}} \Psi(\alpha) = \min_{\vec{\alpha}} \frac{1}{2} \sum_{1=1}^{N} \sum_{j=1}^{N} y_i y_j k\left(\vec{x}_i, \vec{x}_j\right) \alpha_i, \alpha_j - \sum_{i=1}^{N} \alpha \tag{4}$$

Subject to box constraints

$$0 < \alpha_i \leq C, \forall i \tag{5}$$

and one linear equality constraint

$$\sum_{1=j}^{N} y_i \alpha_i = 0 \tag{6}$$

The symbols α_i represent the Lagrange multipliers associated with a primal quadratic programming problem. Each training example $\vec{x}_i$ corresponds uniquely to a Lagrange multiplier. Eqs. (4)–(6) collectively constitute a quadratic programming problem that the SVM algorithm is designed to resolve. The algorithm reaches its conclusion when it fulfills all the QP programming's Karush-Kuhn-Tucker (KKT) optimality conditions of the quadratic programming problem.

$$\begin{aligned} &\alpha_i = 0 \Leftrightarrow y_i u_i \geq 1, \\ &0 < \alpha_i < C \Leftrightarrow y_i u_i = 1, \\ &\alpha_i = C \Leftrightarrow y_i u_i \leq 1 \end{aligned} \tag{7}$$

where u_i is the output of the SVM for the ith training example.

In conclusion, this algorithm makes a significant contribution to the field of change point detection. It excels in both efficiency and accuracy, effectively identifying sequence changepoints by leveraging the (PELT) approach. Its efficient handling of large datasets with a time complexity of O(n) ensures scalability. Moreover, the inclusion of the penalty constant β helps prevent overfitting, enhancing the algorithm's robustness for diverse applications, including anomaly detection, signal processing, and time series analysis. The algorithm's ability to provide valuable insights into the dynamics and trends of the data further enhances its utility in change point detection tasks. Overall, this approach constitutes a valuable and powerful tool for detecting significant changes in time series data in various real-world scenarios.

3.2.4 Testing Set Component

The testing set serves as the evaluation ground for ACIDS-PELT, assessing its performance in terms of accuracy, precision, and recall. It enables a comprehensive understanding of the model's effectiveness in identifying security threats within the cloud environment.

The testing component represents the culminating phase of the system. Following the training process using the SVM algorithm, the testing or evaluation phase assesses incoming network traffic to determine the accuracy of data classification into regular or attack categories. The role and the impact of the component are outlined below:

Role: The testing set is used to evaluate the performance of ACIDS-PELT by measuring its accuracy, precision, and recall in identifying security threats in the cloud environment.

Impact: The testing set serves as a critical assessment tool for the model. It validates the generalizability and effectiveness of ACIDS-PELT in real-world scenarios. The measured accuracy, precision, and recall metrics provide insights into the system's ability to identify and classify security threats correctly.

4 Datasets and Evaluation Metrics

4.1 Datasets

We employed two datasets to assess our proposed Adaptive Cloud Intrusion Detection System (ACIDS-PELT): The widely recognized NSL-KDD dataset and the cloud-based ISOT-CID. This section will offer a concise overview of both datasets.

4.1.1 NSL–KDD Dataset

The selection of the NSL-KDD benchmark dataset for evaluation is justified by its widespread adoption in the research community as a standard benchmark for assessing intrusion detection systems. While acknowledging its limitations, such as being derived from the KDD-Cup 99 dataset, which might not fully represent real-world cloud intrusion scenarios, the NSL-KDD dataset offers a diverse set of network traffic data with labelled instances of normal and malicious activities. This dataset provides a standardized platform for evaluating the proposed Adaptive Cloud Intrusion Detection System (ACIDS-PELT) and allows for comparisons with existing techniques. Additionally, its use aligns with standard practices in the field, enabling researchers to assess the generalizability and effectiveness of the proposed approach within the context of well-established benchmarks. The proposed system will be tested using the NSL-KDD intrusion detection data collection, which is recognized for its realism, diverse attack types, and inclusion of both normal and attack data [43]. An enhanced version of KDD-Cup 99, NSL-KDD, is extensively employed to assess IDS algorithms in conventional networks and cloud computing scenarios. The dataset comprises 41 features with corresponding labels, including records selected from KDD-Cup 99. It features 24 attack types in the training set and introduces 14 new attacks in the test set that are absent in the training data. Table 2 presents the class distribution categorized into DoS, Probe, User to Root (U2R), and Remote to Local (R2L). The training set has 67,343 regular instances, 45,927 DoS instances, 11,656 probe instances, 995 R2L instances, and 52 U2R instances. Meanwhile, the test set comprises 9,711 normal instances, 7,456 DoS instances, 2,421 probe instances, 2,756 R2L instances, and 200 U2R instances [44]. These detailed features provide a comprehensive understanding of the dataset's composition and structure, facilitating a robust evaluation of the proposed Adaptive Cloud Intrusion Detection System (ACIDS). Detailed features of the data can be found in Table 3.

Table 2: The class distribution in ISOT-CID and NSL-KDD datasets

Dataset	ISOT-CID dataset			NSL-KDD dataset		
	Total	Normal	Attack	Total	Normal	Attack
Training	47,657	38,126	9,531	12,5973	67,343	58,630
Testing	11,914	9,531	2,383	22,544	9,711	12,883

Table 3: Features in the NSL-KDD dataset

Feature number	Description	Type	Feature number	Description	Type
1	Duration	Numeric	22	is_guest_login	Numeric
2	Protocol_type	Symbolic	23	count	Numeric
3	Service	Symbolic	24	srv_count	Numeric
4	Flag	Symbolic	25	serror_count	Numeric
5	src_bytes	Numeric	26	srv_serror_rate	Numeric
6	dst_bytes	Numeric	27	rerror_rate	Numeric
7	Land	Numeric	28	srv_error_rate	Numeric
8	Wrong fragment	Numeric	29	same_srv_rate	Numeric
9	Urgent	Numeric	30	diff_srv_rate	Numeric
10	Hot	Numeric	31	srv_diff_host_rate	Numeric
11	num_failed_login	Numeric	32	dst_host_count	Numeric
12	logged_in	Numeric	33	dst_host_srv_count	Numeric
13	num_compromised	Numeric	34	dst_host_same_srv_rate	Numeric
14	root_shell	Numeric	35	dst_host_diff_srv_rate	Numeric
15	su_attempted	Numeric	36	dst_host_same_srv_host_rate	Numeric
16	num_root	Numeric	37	dst_host_srv_diff_host_rate	Numeric
17	num_file_creation	Numeric	38	dst_host_serror_rate	Numeric
18	num_shell	Numeric	39	dst_host_srv_serror_rate	Numeric
19	num_access_file	Numeric	40	dst_host_rerror_rate	Numeric
20	num_out_of_bound_cmd	Numeric	41	dst_host_srv_rerror	numeric
21	is_hot_login	Numeric			

4.1.2 ISOT-CID Dataset

The selection of the ISOT-CID dataset for our cloud intrusion detection research is driven by its authenticity, mirroring real-world cloud environments, and enabling robust evaluation of intrusion detection models. Its substantial size, diverse activities, and multiple intrusion scenarios allow comprehensive testing of detection techniques. The dataset's variety of anomalous activities facilitates the analysis of various threats, while its availability aids comparative evaluations against existing methods. Despite not being explicitly designed for cloud environments, its relevance in modelling network behaviours aligns well with evaluating intrusion detection in modern cloud systems.

ISOT-CID, cited as the primary intrusion dataset publicly accessible, is distinctive for being captured in a genuine cloud environment. The data collection occurred at various layers, including the network, guest host, and hypervisor layers of OpenStack-based cloud nodes [45]. This dataset, gathered in two phases, utilizes the second-phase data for its recent timestamp and coverage of emerging attack patterns. It encompasses diverse attack types, such as simultaneous and coordinated attacks, extending to assaults from multiple continents. Non-malicious records in the dataset represent complex scenarios, reflecting the behaviours of 160 authorized users, encompassing activities like VM maintenance, file creation, SSH usage, updates, and reboots.

The ISOT-CID dataset classifies attack patterns into insider and outsider attacks based on the perpetrator's identity. Insider attacks involve internal users with elevated privileges or compromised VMs that might target other instances in the cloud. Outsider attacks, in contrast, originate from external sources. The dataset, totaling eight terabytes, includes 55.2 GB of network traffic data collected at various internal, external, and local communication levels. Internal traffic pertains to communication between hypervisor nodes, external traffic between different instances, and local traffic between VMs on the same hypervisor node. Network traffic data is in packet capture (PCAP) format, with the first phase capturing 22,372,418 packets (0.07% malicious) and the second phase collecting 11,509,254 packets (17.43% attacks) [46]. Table 2 illustrates the distribution of classes for the ISOT-CID datasets.

4.1.3 Ethical Considerations

When utilizing datasets like NSL-KDD and ISOT-CID for research, ethical considerations regarding data privacy and potential biases are of utmost importance. These datasets often contain sensitive information, raising user privacy and confidentiality concerns. Researchers must exercise caution and ensure the anonymization or de-identification of personally identifiable information to mitigate the risk of data breaches or privacy infringements. Additionally, inherent biases within these datasets, such as uneven representation of certain classes or overrepresentation of specific types of attacks, could influence the performance and generalization of intrusion detection models. Acknowledging and addressing these biases is crucial to prevent skewed outcomes and ensure the fairness and reliability of the research findings. Adopting transparent methodologies, maintaining data anonymization, and actively identifying and mitigating biases are imperative ethical practices when working with sensitive datasets like NSL-KDD and ISOT-CID.

4.1.4 Experimental Setup

The PELT change point detection algorithm utilizes two main parameters: The negative log-likelihood as the cost function and Akaike's Information Criterion (AIC) as the penalty. These parameters are widely used in change point detection for evaluating model fit and balancing model complexity. However, their universal effectiveness across different datasets may vary, potentially leading to missed change points or false detections in specific scenarios. Careful parameter selection is crucial to optimize performance based on dataset characteristics and analytical context [36].

4.2 Evaluation Metrics

Various metrics are utilized to gauge a classifier's performance, encompassing True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN), and overall accuracy, pivotal in machine learning and intrusion detection systems (IDS) evaluations. TP, which denotes correctly identified genuine intrusions, emphasizes ACIDS-PELT's sensitivity to actual threats, protecting the cloud environment from potential attacks. A high TP rate ensures reduced risks and safeguards sensitive data. TN, representing accurately labelled normal events, reflects ACIDS-PELT's specificity, reducing unnecessary alerts and ensuring efficient resource allocation. High TN minimizes disruptions to normal operations and alleviates operational strain. FP, denoting normal events misclassified as intrusions, highlights the system's potential for false alarms, impacting resource utilization and operational efficiency. Low FP is crucial for reducing unnecessary investigations and maintaining alert trustworthiness. FN, illustrating undetected genuine intrusions, uncovers ACIDS-PELT's vulnerabilities, which are essential to fortify security and prevent exploitation. Overall accuracy, though a comprehensive measure, can mislead in imbalanced datasets. To assess ACIDS-PELT, its high TP, and low FN rates prove its effectiveness in detecting real threats, fortifying cloud security, and reducing potential damage. A low FP rate optimizes resource allocation, allowing focus on genuine threats. A high TN rate signifies normalcy preservation, reducing operational fatigue and nurturing trust. A comprehensive understanding of ACIDS-PELT's strengths and weaknesses through these metrics directs targeted enhancements and optimized performance.

4.3 Classification Accuracy

The classification accuracy of an algorithm is determined by calculating the percentage of correctly classified instances among the total instances in a dataset. It is represented using Eq. (8).

$$Accuracy = \frac{TN + TP}{TN + FP + FN + TP} \times 100 \tag{8}$$

i. Detection Rate

The detection rate, often referred to as the true positive rate, signifies the proportion of malicious traffic that is accurately identified or detected by the system. It can be calculated using Eq. (9).

$$Detection\ Rate = \frac{TP}{TP + FN} \times 100 \tag{9}$$

ii. False Positive Rate

The false-positive rate (FPR) represents the ratio or percentage of normal traces or benign activities that are inaccurately identified as attacks or anomalies. It can be calculated using Eq. (10).

$$FPR = \frac{FP}{TN + FP} \times 100 \tag{10}$$

5 Experimental Results

We utilized the Jupiter Notebook and Python library to construct the system. We employed two datasets to assess the model's performance: ISOT-CID and NSL-KDD. The ISOT-CID dataset, the first publicly accessible cloud intrusion dataset, contains raw data gathered from diverse components of real-world cloud infrastructure on the OpenStack platform. Conversely, the NSL-KDD dataset serves as a well-known IDS benchmark originating from a traditional network environment. The ISOT-CID is split into two datasets: D1, consisting of 24,707 instances with diverse activities like normal network behavior, network scan attacks, and DDoS attacks, subdivided into a 60% training set and a 40% test set. Each dataset was divided into six equal samples, allowing observation of data evolution. D2 comprised 43,490 instances with similar activities and was meticulously divided into training and test sets, enabling a comprehensive assessment of the IDS's performance across varied conditions ensuring its real-world robustness.

In addition to the SVM configuration, we integrated the PELT change point detection algorithm. This algorithm boasts two critical parameters: The cost function and the penalty. In our implementation, the cost function was set as negative log-likelihood. At the same time, the penalty was based on AIC (Akaike Information Criterion), which finds a balance between goodness of fit and model complexity by leveraging information theory. AIC favors models that fit the data well while penalizing more complex models to prevent overfitting. These choices align with standard practices for change point detection [36].

5.1 Feature Selection Performance

After the feature selection process, 13 features were chosen from the original 41 in the NSL-KDD dataset. The ISOT-CID dataset had 390 features before preprocessing, which were later reduced to 186. Subsequently, 17 significant features were selected for analysis. This reduction emphasizes the focus on pertinent attributes for intrusion detection. The selected features from both datasets, listed in Table 4, encompass crucial attributes, including network service, communication parameters, packet sizes, login status, and server-related metrics. Carefully chosen for their relevance in change point detection scenarios, they capture various network communication facets like inter-arrival times, TCP/IP protocol parameters, and payload characteristics. Their importance lies in detecting deviations from standard network behaviour and identifying potential intrusions or anomalies. Table 4 underscores their significance in identifying alterations associated with security threats.

Table 4: Selected features from NSL-KDD and ISOT-CID datasets

Datasets	Selected features	Number of features selected
NSL-KDD	Service, flag, src_bytes, dst_bytes, logged_in, count, serror_count, same_srv_rate, diff_srv_rate, dst_host_srv_count, dst_host_same_srv_rate, dst_host_serror_rate, dst_host_srv_serror_rate	13
ISOT-CID	dsLowQuartileIat, dsModeIat, tp0fDis, tcpWinSzThRt, tcpEcI, fnHash, connSipDip, PyldChRatio, stdIAT, connF, tcpTmS, tcpTmER,flowInd, day, hour, minute, second	17

5.2 The Change Point Detection Results

Cloud Intrusion Detection System (IDS) scalability is a critical issue the suggested solution attempts to address. As depicted in Table 5, the change point analysis carried out across multiple scenarios involving ISOT-CID datasets D1 and D2 revealed changes in statistical properties, particularly in mean, at multiple locations. The ISOT-CID dataset was utilized for this analysis, specifically focusing on the first scenario where the dataset was split. Employing the (PELT) change point algorithm, the study aimed to detect alterations in statistical properties, particularly the mean, across Samples 1 to 6. The figures presented in the analysis display red horizontal lines denoting the identified change points in the mean values. These figures illustrate distinctive shifts in the mean values at different sample locations. A thorough analysis of these observed changes holds significant implications for intrusion detection in cloud environments, as detecting alterations in statistical properties, such as mean values, suggests potential anomalies or shifts in network behaviour, signifying critical points where intrusion attempts or abnormal activities are initiated. Understanding these variations enhances intrusion detection mechanisms in cloud environments, fortifying security measures against potential threats or attacks.

Table 5: Result of change point on all scenarios of D1 and D2

Scenario	Training set size & position	Testing set size & position
1	1000 (1–1000)	2000 (1001–3000)
2	2000 (1–2000)	2000 (2001–4000)
3	3000 (1–1000)	2000 (3001–5000)
4	2000 (1–2000)	2000 (4001–6000)
5	1000 (1–1000)	2000 (5001–7000)
6	3000 (1–3000)	2000 (6001–8000)

Moreover, by pinpointing these change points, security systems can adapt proactively, improving their responsiveness and resilience against emerging security risks within cloud networks. In Scenario 1 and 4, four change points occurred, identified respectively at the 850th, 950th, 1300th, and 3000th instances. Conversely, Scenarios 2 to 6 demonstrated two change points at the 1300th and 3000th instances, highlighted in Table 5 and Fig. 2.

The dynamic nature of the cloud environment is intimately related to these changes in statistical properties. For example, changes in the mean caused by the cloud infrastructure's scalability of virtual machines (VMs) can cause false alarms in the IDS. As such, updating the IDS Reference Model as soon as any changes are detected is necessary to reduce the likelihood of false alarms. Interestingly, the average time before a change in a statistical property of the data happens is reflected in the mean interval between consecutive change points. Using this realization, the Support Vector Machine (SVM) classification algorithm's reference model is refined during a crucial update period represented by the mean interval between these change points. This tactical method guarantees that the intrusion detection system (IDS) will continue to adapt to the changing features of the cloud environment, thereby reducing false alarms and preserving strong intrusion detection capabilities.

Due to the dynamic nature of cloud environments, we have noticed variations in the means between the different instances in the scenarios. In particular, the VM scaling fluctuations can potentially cause false alarms to be raised by the Intrusion Detection System (IDS). Thus, to reduce the number of false alarms, the IDS Reference Model must be updated as soon as such changes are discovered. Knowing the average time between successive change points becomes crucial because it indicates the average time before a change appears in the data's statistical property. Using this data, the SVM classification algorithm's reference model is updated regularly using the mean interval between subsequent change points.

In exploring the NSL-KDD dataset through distinct training and testing set configurations, change point analysis unveiled varying statistical properties indicative of potential shifts in network traffic characteristics. The scenario with a 1000-instance training set (1–1000) and subsequent 2000-instance testing set (1001–3000) revealed distinct alterations, implying changes in distribution or trends within the network data. Similarly, in the scenario utilizing a 2000-instance training set (1–2000) and a subsequent 2000-instance testing set (2001–4000), the analysis detected shifts in statistical properties, hinting at evolving patterns or anomalies in network traffic behaviour as shown in Table 6. These findings underscore the dynamic nature of the NSL-KDD dataset and the potential for evolving network patterns captured by different training and testing set configurations.

Table 6: Classification outcomes for NSL-KDD datasets

Dataset/Scenario	Training set size & position	Testing set size & position
NSL-KDD	1000 (1–1000)	2000 (1001–3000)
NSL-KDD	2000 (1–2000)	2000 (2001–4000)

Table 7 presents the comparison between the scenario before scaling, displaying an accuracy of 99.1%, and after scaling, registering an accuracy of 98.8%, which offers critical insights into the ISOT-CID dataset's model performance before and after VM scaling. Initially, pre-scaling exhibited a robust model performance, showcasing a high Detection Rate (DR) of 99.7% for Port scan and 97.1% for DDoS, along with a noticeable increase in False Positive Rate (FPR) at 0.9% for Port scan and 2.8% for DDoS. However, post-scaling, despite the overall improvement in accuracy to 99.1%, there was a marginal decline in specific metrics. The model showed a further enhancement in DR, recording 99.1% for Port scan and 96.9% for DDoS after scaling. Nonetheless, a slight increase in FPR was observed, slightly elevating to 1.3% for Port scan and 2.9% for DDoS post-scaling. Notably, this slight performance degradation post-scaling, particularly in the FPR metrics, may be attributed to scalability issues. These findings suggest an overall improvement in the model's adaptability post-scaling, albeit with a marginal compromise in specific performance metrics due to scalability concerns, highlighting the model's enhanced intrusion detection capabilities in dynamic environments.

Table 7: Classification outcomes for ISOT-CID dataset

Data scenario	Before V scaling					After VM scaling				
	D1					D2				
	In overall accuracy (%)	DR (%)		FPR (%)		In overall accuracy (%)	DR (%)		FPR (%)	
		Port scan	DDoS	Port scan	DDoS		Port scan	DDoS	Port scan	DDoS
1	98.4	99.6	97.3	1.2	2.5	98.8	98.9	95.9	1.5	2.7
2	99.1	98.9	98.4	0.9	3.1	99.0	99.2	96.7	1.1	3.2
3	98.8	100	96.5	0.3	2.8	97.4	98.8	97.5	1.3	2.8
4	99.9	99.8	97.3	0.2	3.5	98.2	99.6	97.8	0.8	3.5
5	99.5	99.7	96.8	1.5	2.7	99.5	98.9	97.1	1.7	2.7
6	98.9	99.9	96.5	1.3	2.2	99.7	98.9	96.5	1.5	2.3
Average	**99.1**	**99.7**	**97.1**	**0.9**	**2.8**	**98.8**	**99.1**	**96.9**	**1.3**	**2.9**

In contrast to the variability observed in the cloud-based dataset scenarios, the NSL-KDD dataset, as shown in Table 8, showcases an exemplary model performance. With an exceptional overall accuracy of 99.99%, the NSL-KDD dataset outperforms the cloud-based scenarios in accuracy measures. Similarly, boasting a Detection Rate (DR) of 98.98% and an incredibly low False Positive Rate (FPR) of 0.01%, the NSL-KDD dataset underscores a notably higher precision in detecting network intrusions compared to the fluctuating performance metrics identified in the cloud-related data scenarios. This stark contrast highlights the consistency and reliability of the NSL-KDD dataset's model performance, signifying its potential as a benchmark or robust foundation for intrusion detection systems in network security applications, especially when compared to the observed variations in the cloud-based dataset's model outcomes.

Table 8: Classification outcomes for NSL-KDD dataset

Dataset scenario	Accuracy (%)	True positive (%)	False positive (%)
1	99.98	98.97	0.01
2	99.99	98.98	0.02
3	99.99	98.98	0.01
Average	**99.99**	**98.98**	**0.01**

6 Comparison with Existing Techniques

The proposed adaptive system, ACIDS-PELT, underwent a comparative analysis against non-adaptive anomaly detection techniques like K-Means and Random Forest as proposed by K. Samunnisa, alongside two adaptive systems suggested by Jamal TALBI and Bikash Agrawal in their respective works addressing scalability issues in anomaly detection. This extensive evaluation, detailed in Tables 9, 10 and Figs. 4 to 7, revealed insightful performance differences among the compared approaches. Let us delve deeper into the strengths and weaknesses of each approach to understand why ACIDS-PELT shines.

Table 9: Comparative evaluation of ACIDS-PELT against existing techniques using NSL-KDD

Dataset	Accuracy				Detection rate				False positive rate			
	ACIDS-PELT	K-Means	Random forest	ACA-TS-IaaS-CD	ACIDS-PELT	K-Means	Random forest	ACA-TS-IaaS-CD	ACIDS-PELT	K-Means	Random forest	ACA-TS-IaaS-CD
NSL-KDD	99.99	75.5	78.4	87	98.98	72.4	76.8	88	0.01	20.9	19.4	14

Table 10: Assessing ACIDS-PELT performance: Comparative analysis with existing techniques utilizing ISOT-CID dataset

IDS techniques	Before VM scaling					After VM scaling				
	D1					D2				
	Overall accuracy (%)	DR (%)		FPR (%)		Overall accuracy (%)	DR (%)		FPR (%)	
		Port scan	DDoS	Port scan	DDoS		Port scan	DDoS	Port scan	DDoS
ACIDS-PELT	**99.1**	**99.7**	**97.1**	**0.9**	**2.8**	**98.8**	**99.1**	**96.9**	**1.3**	**2.9**
K-Means	89.3	90.3	86.9	9.2	15.2	85.9	90.5	83.5	16.6	16.2

(Continued)

Table 10 (continued)

IDS techniques	Before VM scaling					After VM scaling				
	D1					D2				
	Overall accuracy (%)	DR (%)		FPR (%)		Overall accuracy (%)	DR (%)		FPR (%)	
		Port scan	DDoS	Port scan	DDoS		Port scan	DDoS	Port scan	DDoS
Random forest	92.9	91.3	90.2	7.5	8.2	90.4	92.7	87.7	8.9	13.2
ACA-TS-IaaS-CD	88.5	94.2	92.5	6.8	10.5	85.6	92.9	87.9	8.6	14.9

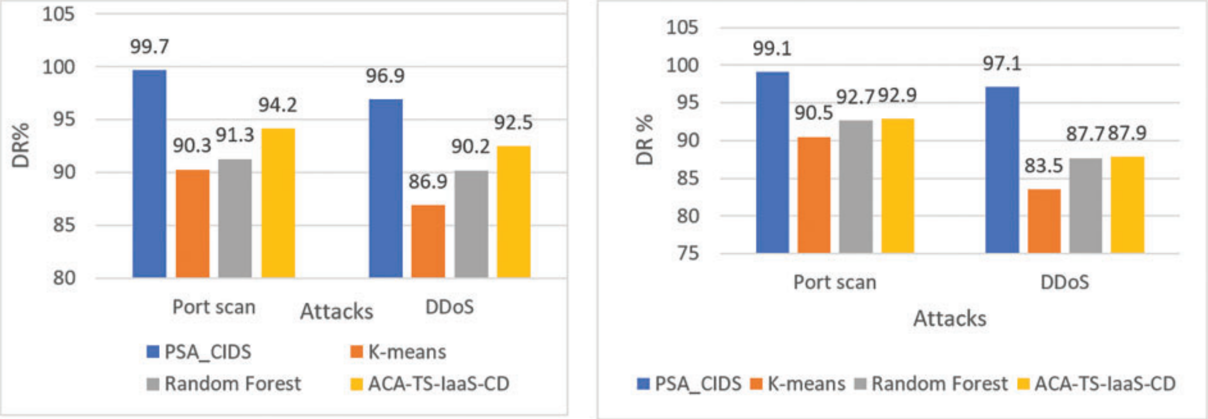


Figure 4: Comparing the detection rates of CIDS-PELT and current methods on the ISOT-CID dataset before and after VM scaling

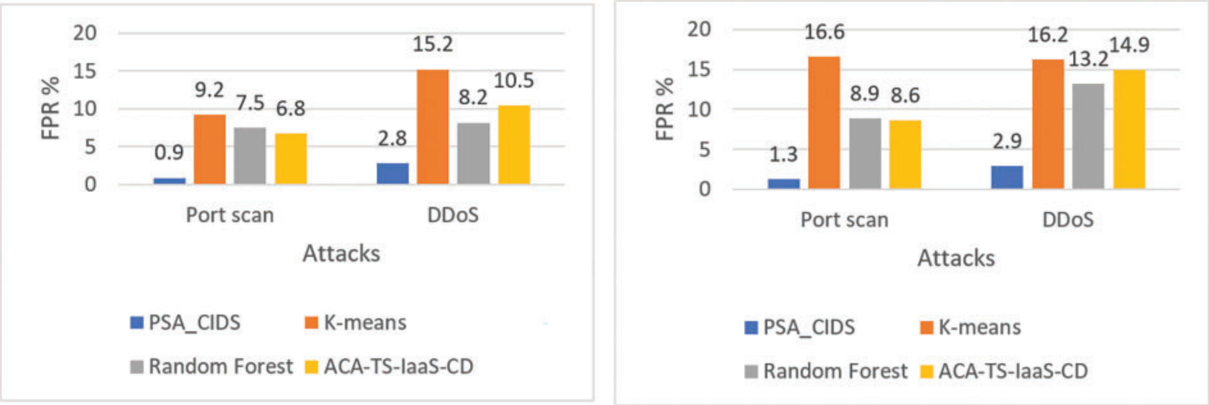


Figure 5: Comparing the false positive rate of CIDS-PELT and current methods on the ISOT-CID dataset before and after VM scaling

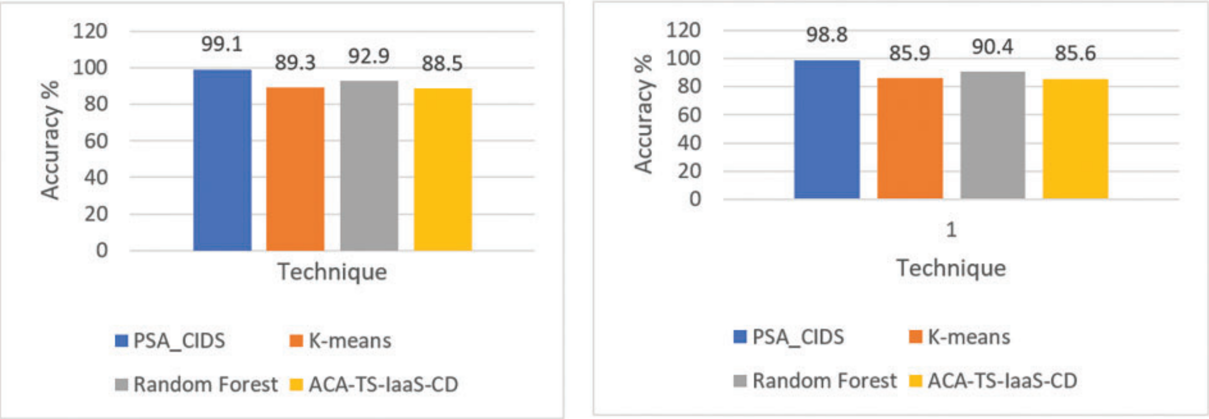


Figure 6: Comparing the accuracy of CIDS-PELT and current methods on the ISOT-CID dataset before and after VM scaling

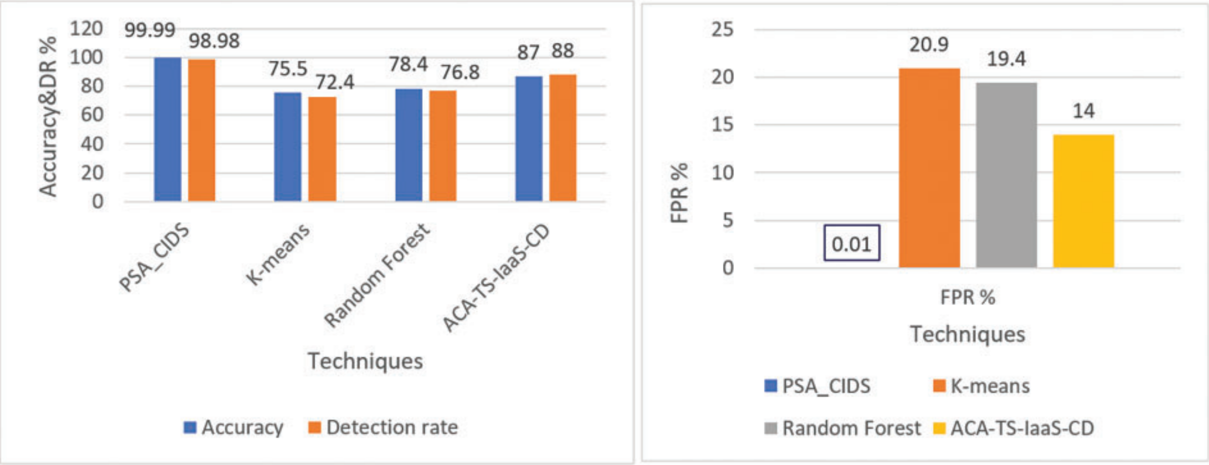


Figure 7: Comparing the accuracy & DR and FPR of CIDS-PELT and current methods on NSL-KDD dataset before and after VM scaling

Non-adaptive Approaches:

- K-Means: While its simplicity offers ease of implementation, K-Means struggles with dynamic, evolving cloud environments. Its static clusters fail to adapt to new attack patterns, leading to lower detection rates, especially for novel threats like DDoS (88.5% in D1). Its high false positive rate (19.7% for DDoS in D1) also creates unnecessary resource overhead and alerts.
- Random Forest: While offering better flexibility than K-Means, Random Forest's black-box nature makes it challenging to interpret and fine-tune for specific attack types. Its accuracy (92.9% in D1) suffers compared to ACIDS-PELT, and its relatively high false positive rate (8.6% for DDoS in D1) can still create resource strain.

Adaptive Approach-A-CA-TS-IaaS-CD:

- This approach attempts to adapt to changing workload dynamics, but its reliance on resource thresholds makes it susceptible to misinterpreting normal fluctuations as anomalies. This leads to lower detection rates (85.6% in D2) and higher false positive rates (14.4% for DDoS in D2) than ACIDS-PELT. Additionally, its complex architecture necessitates significant computational resources, limiting its scalability.

ACIDS-PELT: The Adaptive Edge

ACIDS-PELT stands out through its combination of strengths:

- Superior Accuracy and Detection Rates: Across both D1 and D2, ACIDS-PELT consistently delivers the highest overall accuracy (99.1% in D1, 98.8% in D2) and excels in detecting specific threats like port scans (99.7% in D1, 99.1% in D2) and DDoS (97.1% in D1, 96.9% in D2). This underscores its effectiveness in identifying both known and novel attacks.
- Minimized False Positives: ACIDS-PELT's false positive rates are substantially lower than other approaches (0.9% for port scans, 2.9% for DDoS in D1, 1.3% for port scans, 2.9% for DDoS in D2). This reduces unnecessary resource consumption and operational fatigue caused by false alarms.
- Adaptability and Scalability: ACIDS-PELT's dynamic adjustment capabilities enable it to handle changing workloads and maintain high performance even after VM scaling (D2). Additionally, its lightweight architecture minimizes resource requirements, making it suitable for large-scale cloud deployments.

In conclusion, the comprehensive comparison confirms ACIDS-PELT's compelling advantages in accuracy, detection rate, and efficiency over both non-adaptive and adaptive approaches. Its unique combination of adaptability and scalability positions it as a powerful solution for addressing anomaly detection challenges in dynamic cloud environments.

The effectiveness of ACIDS-PELT was rigorously validated using two well-known datasets: The ISOT-CID cloud dataset and the NSL-KDD intrusion detection benchmark. Tables 9, 10 and Figs. 4–7 present comprehensive results comparing ACIDS-PELT's accuracy, detection rate, and false positive rate to K-Means, Random Forest, and A-CA-TS-IaaS-CD.

The analysis provides a convincing and detailed picture of ACIDS-PELT. Fig. 4 shows how it consistently outperforms the compared methods regarding detection rate. Fig. 5 demonstrates its advantage by displaying significantly lower false positive rates. Figs. 6 and 7 demonstrate ACIDS-PELT's unrivalled accuracy and false positive rate performance across the board. The NSL-KDD dataset analysis reveals similar trends of increased efficacy and improved performance metrics for ACIDS-PELT, extending these compelling results beyond ISOT-CID. Finally, the validation tests show that ACIDS-PELT has the potential to revolutionize cloud intrusion detection. Its superior performance across various datasets and metrics makes it a strong contender for real-world implementation, providing a robust and dependable defense against evolving cyber threats.

7 Discussion

The classification outcomes obtained from the ISOT-CID dataset for the proposed Adaptive Cloud Intrusion Detection System (ACIDS) using PELT and SVM exhibit a comprehensive performance evaluation. The system's adaptability to diverse scenarios, both before and after VM scaling, is evident in the results. Before VM scaling, the model achieved an overall accuracy averaging 99.1%, with a detection rate (DR) of 99.7% and a false positive rate (FPR) of 0.9%. The system consistently identified port scan and DDoS attacks, showcasing DRs exceeding 98.9% across multiple scenarios. After VM scaling, the ACIDS model maintained a commendable overall accuracy of 98.8%, demonstrating the system's resilience to changes in the cloud environment. Despite a slight increase in the FPR to 1.3%, the model sustained robust performance in detecting different attack types, especially DDoS attacks, emphasizing its effectiveness and reliability. These results underscore the proposed ACIDS-PELT-SVM system's capacity to effectively adapt to varying conditions in cloud environments while ensuring high accuracy in detecting potential intrusion attempts.

In Table 8, we present the noteworthy results achieved by the proposed Adaptive Cloud Intrusion Detection System (CIDS-PELT) in terms of accuracy, detection rate, and false-positive rate. Remarkably, our proposed system achieved an exceptional average accuracy of 99.99%, demonstrating its robustness in accurately identifying intrusions. Furthermore, the True Positive Rate also stood at an impressive 99.99%, indicating its capability to effectively detect genuine intrusion attempts. Importantly, the system exhibited an exceptionally low False Positive Rate of just 0.01%, highlighting its ability to significantly reduce false alarms.

To provide a comprehensive perspective, we conducted a comparative analysis by juxtaposing the performance of CIDS-PELT with that of our previous non-adaptive cloud IDS and an adaptive anomaly detection system tailored for the cloud, as documented in prior research. This detailed comparison, depicted in Tables 9, 10 and Figs. 4–7, illuminates the distinct advantages of CIDS-PELT.

It has been noticed that the statistical attributes of cloud data exhibit fluctuations at various intervals, as indicated by the red vertical lines in Fig. 2. These alterations underscore the dynamic nature of cloud data, necessitating adaptive detection mechanisms to accommodate these evolving data behaviours. Consequently, the Adaptive Gradient Algorithm (SVM) was implemented to continuously update the model parameters in accordance with these change patterns. Results obtained from this proposed system, illustrated in Tables 7, 8 and Figs. 4–7, demonstrate that the adaptive system showcased superior performance in both pre-and post-VM scaling compared to existing techniques. The scaling of Virtual Machines (VMs) adversely impacts the performance of cloud IDS by inducing a dynamic and unstable environment, thereby posing challenges to anomaly detection. To address these shifts in data behaviour, the proposed technique incorporates adaptability to update the normal reference model. Specifically, observations revealed a decline in performance for K-Means, Random Forest, and A-CA-TS-IaaS-CD after scaling scenarios, likely attributable to the static nature of these techniques.

The real-world application of ACIDS-PELT in operational cloud environments presents promising prospects and potential challenges. Its adaptability and accuracy in detecting anomalies are crucial for ensuring cloud security. However, deploying ACIDS-PELT in operational cloud environments might face computational efficiency and scalability hurdles. Real-world cloud systems handle extensive data traffic, requiring robust systems capable of efficient processing and scalability to adapt to varying workloads without compromising accuracy. Implementing ACIDS-PELT might pose resource consumption and time complexity challenges, particularly in large-scale cloud infrastructures. Balancing high detection accuracy with computational efficiency is essential to ensure practical deployment in operational cloud environments. Addressing these challenges will be vital for integrating ACIDS-PELT effectively into real-world cloud environments and maximizing its potential benefits for enhanced security measures.

8 Conclusion

This paper significantly contributes to change point detection by introducing an efficient algorithm based on the (PELT) approach. The algorithm is scalable with a penalty constant (β) and excels in anomaly detection and signal processing applications. Because of its ability to provide valuable insights into data dynamics, it is more helpful in detecting significant changes in time series data in real-world scenarios. The paper also introduces ACIDS-PELT, an adaptive intrusion detection system tailored for the cloud environment. The system selects relevant features, monitors thresholds for retraining, and performs practical training and detection using four key components. ACIDS-PELT showcases superiority through testing on the ISOT-CID and NSL-KDD dataset, surpassing related works in intrusion detection.

ACIDS-PELT strategically addresses challenges in dynamic and distributed cloud environments, ensuring dynamic adaptability with the integration of PELT. Its benefits include dynamic adaptability, precise anomaly detection, and superior intrusion detection. Evaluations with the ISOT-CID and NSL-KDD datasets demonstrate its effectiveness, addressing challenges related to approximation and imprecision.

While ACIDS-PELT shines in precise change detection, its real-world effectiveness requires further polish:

1). Compute Crunch: Optimizing ACIDS-PELT's resource usage is crucial for smooth operation in resource-constrained cloud environments.

2). False Alarm Fatigue: The system needs stricter validation to ensure advanced scoring minimizes false positives without jeopardizing threat detection.

3). Evolving Foes: Self-learning and threat intel integration sound good, but their effectiveness against real-world evolving threats needs concrete testing.

4). Hybrid Helpers or Redundant Relics: Exploring hybrid approaches holds promise, but complexity and cost-benefit analysis are essential to avoid unnecessary baggage.

5). Counting the Coins: A thorough cost-benefit analysis considering deployment costs, resource savings, and overall security impact is vital for real-world deployment decisions. By addressing these limitations, ACIDS-PELT can transform from a promising concept into a robust and cost-effective guardian for the ever-evolving cloud landscape.

Acknowledgement: The authors extend their appreciation to the Deanship of Scientific Research at Imam Mohammad Ibn Saud Islamic University (IMSIU) for funding and supporting this work.

Funding Statement: This work is funded by the Deanship of Scientific Research at Imam Mohammad Ibn Saud Islamic University (IMSIU) through Research Partnership Program No. RP-21-07-09.

Author Contributions: The authors confirm their contribution to the paper as follows: Study conception and design: W. Elabkri, M. Md. Siraj, B.A.S. Al-rimy, S.N. Qasem, T. Al-Hadhrami; data collection: W. Elabkri, M. Md. Siraj, B.A.S. Al-rimy; analysis and interpretation of results: W. Elabkri, M. Md. Siraj, B.A.S. Al-rimy, S.N. Qasem, T. Al-Hadhrami; draft manuscript preparation: W. Elabkri, M. Md. Siraj, B.A.S. Al-rimy. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Data and materials are accessible upon request and are publicly available.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] P. Mell and T. Grance, "The NIST definition of cloud computing," *Commun ACM*, 2011. doi: 10.6028/NIST.SP.800-145.

[2] M. T. Khorshed, A. S. Ali, and S. A. Wasimi, "A survey on gaps, threat remediation challenges and some thoughts for proactive attack detection in cloud computing," *Future Gener. Comput. Syst.*, vol. 28, no. 6, pp. 833–851, 2012. doi: 10.1016/j.future.2012.01.006.

[3] K. Sethi, R. Kumar, D. Mohanty, and P. Bera, "Robust adaptive cloud intrusion detection system using advanced deep reinforcement learning," in *Int. Conf. Secur., Priv., Appl. Cryptogr. Eng.*, Kolkata, India, Springer, Dec. 17–21, 2020, pp. 66–85.

[4] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel and M. Rajarajan, "A survey of intrusion detection techniques in cloud," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 42–57, 2013. doi: 10.1016/j.jnca.2012.05.003.

[5] F. Amiri, M. R. Yousefi, C. Lucas, A. Shakery, and N. Yazdani, "Mutual information-based feature selection for intrusion detection systems," *J. Netw. Comput. Appl.*, vol. 34, no. 4, pp. 1184–1199, 2011. doi: 10.1016/j.jnca.2011.01.002.

[6] H. J. Liao, C. H. R. Lin, Y. C. Lin, and K. Y. Tung, "Intrusion detection system: A comprehensive review," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 16–24, 2013. doi: 10.1016/j.jnca.2012.09.004.

[7] C. F. Tsai, Y. F. Hsu, C. Y. Lin, and W. Y. Lin, "Intrusion detection by machine learning: A review," *Expert Syst. Appl.*, vol. 36, no. 10, pp. 11994–12000, 2009. doi: 10.1016/j.eswa.2009.05.029.

[8] D. Krishnan and M. Chatterjee, "An adaptive distributed intrusion detection system for cloud computing framework," in *Int. Conf. Secur. Comput. Netw. Distrib. Syst.*, Trivandrum, India, Springer, Oct. 11–12, 2012, pp. 466–473.

[9] M. Idhammad, K. Afdel, and M. Belouch, "Distributed intrusion detection system for cloud environments based on data mining techniques," *Procedia Comput. Sci.*, vol. 127, pp. 35–41, 2018. doi: 10.1016/j.procs.2018.01.095.

[10] N. M. Ibrahim and A. Zainal, "An adaptive intrusion detection scheme for cloud computing," *Int. J. Swarm Intell. Res. (IJSIR)*, vol. 10, no. 4, pp. 53–70, 2019. doi: 10.4018/IJSIR.

[11] S. Hassan and F. Azam, "Analysis of cloud computing performance, scalability, availability, & security," in *2014 Int. Conf. Inf. Sci. Appl. (ICISA)*, 2014, pp. 1–5.

[12] T. S. Somasundaram, V. Prabha, and M. Arumugam, "Scalability issues in cloud computing," in *2012 Fourth Int. Conf. Adv. Comput. (ICoAC)*, IEEE, 2012, pp. 1–5.

[13] J. Jiang *et al.*, "AERF: Adaptive ensemble random fuzzy algorithm for anomaly detection in cloud computing," *Comput. Commun.*, vol. 200, pp. 86–94, 2023. doi: 10.1016/j.comcom.2023.01.004.

[14] S. Maiti, C. Garai, and R. Dasgupta, "A detection mechanism of DoS attack using adaptive NSA algorithm in cloud environment," in *2015 Int. Conf. Comput., Commun. Secur. (ICCCS)*, IEEE, 2015, pp. 1–7.

[15] A. Carlin, M. Hammoudeh, and O. Aldabbas, "Defence for distributed denial of service attacks in cloud computing," *Procedia Comput. Sci.*, vol. 73, pp. 490–497, 2015. doi: 10.1016/j.procs.2015.12.037.

[16] S. Alam, M. Shuaib, and A. Samad, "A collaborative study of intrusion detection and prevention techniques in cloud computing," in *Int. Conf. Innov. Comput. Commun.*, Springer Singapore, Springer, 2019, vol. 1, pp. 231–240.

[17] Z. Li, W. Sun, and L. Wang, "A neural network based distributed intrusion detection system on cloud platform," in *2012 IEEE 2nd Int. Conf. Cloud Comput. Intell. Syst.*, IEEE, 2012, vol. 1, pp. 75–79. doi: 10.1109/CCIS.2012.6664371.

[18] C. Huang, G. Min, Y. Wu, Y. Ying, K. Pei and Z. Xiang, "Time series anomaly detection for trustworthy services in cloud computing systems," *IEEE Trans. Big Data*, vol. 8, no. 1, pp. 60–72, 2017.

[19] P. Nagarajan and G. Perumal, "A neuro fuzzy based intrusion detection system for a cloud data center using adaptive learning," *Cybern. Inf. Technol.*, vol. 15, no. 3, pp. 88–103, 2015. doi: 10.1515/cait-2015-0043.

[20] H. H. Chou and S. D. Wang, "An adaptive network intrusion detection approach for the cloud environment," in *2015 Int. Carnahan Conf. Secur. Technol. (iCCST)*, IEEE, 2015, pp. 1–6.

[21] Q. Xia, T. Chen, and W. Xu, "CIDS: Adapting legacy intrusion detection systems to the cloud with hybrid sampling," in *2016 IEEE Int. Conf. Comput. Inf. Technol. (CIT)*, IEEE, 2016, pp. 508–515.

[22] O. A. Wahab, J. Bentahar, H. Otrok, and A. Mourad, "I know you are watching me: Stackelberg-based adaptive intrusion detection strategy for insider attacks in the cloud," in *2017 IEEE Int. Conf. Web Services (ICWS)*, IEEE, 2017, pp. 728–735.

[23] W. Meng, Y. Wang, W. Li, Z. Liu, J. Li and C. W. Probst, "Enhancing intelligent alarm reduction for distributed intrusion detection systems via edge computing," in *23rd Australasian Conf., ACISP 2018*, Wollongong, NSW, Australia, Springer, Jul. 11–13, 2018, vol. 10946, pp. 759–767.

[24] P. Sharma, J. Sengupta, and P. Suri, "WLI-FCM and artificial neural network based cloud intrusion detection system," *Int. J. Adv. Netw. Appl.*, vol. 10, no. 1, pp. 3698–3703, 2018. doi: 10.35444/IJANA.2018.10014.

[25] M. A. Hatef, V. Shaker, M. R. Jabbarpour, J. Jung, and H. Zarrabi, "HIDCC: A hybrid intrusion detection approach in cloud computing," *Concurr. Comput.*, vol. 30, no. 3, pp. e4171, 2018. doi: 10.1002/cpe.4171.

[26] M. M. Sakr, M. A. Tawfeeq, and A. B. El-Sisi, "Network intrusion detection system based PSO-SVM for cloud computing," *Int. J. Comput. Netw. Inf. Secur.*, vol. 11, no. 3, pp. 22–29, 2019. doi: 10.5815/ijcnis.2019.03.04.

[27] P. S. Deshpande, S. C. Sharma, S. K. Peddoju, P. S. Deshpande, S. C. Sharma and S. K. Peddoju, "A network-based intrusion detection system," in *Security Data Storage Aspect Cloud Comput.*, 2019, vol. 52, pp. 35–48.

[28] Z. Chiba, N. Abghour, K. Moussaid, A. El Omri, and M. Rida, "A clever approach to develop an efficient deep neural network based IDS for cloud environments using a self-adaptive genetic algorithm," in *2019 Int. Conf. Adv. Commun. Technol. Netw. (CommNet)*, IEEE, 2019, pp. 1–9.

[29] Z. Chiba, M. S. E. K. Alaoui, N. Abghour, and K. Moussaid, "Automatic building of a powerful IDS for the cloud based on deep neural network by using a novel combination of simulated annealing algorithm and improved self-adaptive genetic algorithm," *Int. J. Commun. Netw. Inf. Secur.*, vol. 14, no. 1, pp. 93–117, 2022. doi: 10.17762/ijcnis.v14i1.5264.

[30] Z. Q. Liu, B. Xu, B. Cheng, X. M. Hu, and M. Darbandi, "Intrusion detection systems in the cloud computing: A comprehensive and deep literature review," *Concurr. Comput.-Pract. Exp.*, vol. 34, no. 4, pp. e6646, 2022. doi: 10.1002/cpe.6646.

[31] X. Jin and X. Qiu, "An adaptive anomaly detection method for cloud computing system," in *2022 IEEE 5th Int. Conf. Electron. Technol. (ICET)*, IEEE, 2022, pp. 1289–1295.

[32] K. Samunnisa, G. S. V. Kumar, and K. Madhavi, "Intrusion detection system in distributed cloud computing: Hybrid clustering and classification methods," *Measure.: Sens.*, vol. 25, pp. 100612, 2023.

[33] N. Sarkar, P. K. Keserwani, and M. C. Govil, "A better and fast cloud intrusion detection system using improved squirrel search algorithm and modified deep belief network," *Cluster Comput.*, vol. 27, pp. 1699–1718, 2023.

[34] L. K. Vashishtha, A. P. Singh, and K. Chatterjee, "HIDM: A hybrid intrusion detection model for cloud based systems," *Wirel. Pers. Commun.*, vol. 128, no. 4, pp. 2637–2666, 2023. doi: 10.1007/s11277-022-10063-y.

[35] G. Nagarajan and P. Sajith, "Optimization of BPN parameters using PSO for intrusion detection in cloud environment," *Soft Comput.*, vol. 23, pp. 1–12, 2023. doi: 10.1007/s00500-023-08737-1.

[36] R. Killick, P. Fearnhead, and I. A. Eckley, "Optimal detection of changepoints with a linear computational cost," *J. Am. Stat. Assoc.*, vol. 107, no. 500, pp. 1590–1598, 2012. doi: 10.1080/01621459.2012.737745.

[37] M. H. Aghdam and P. Kabiri, "Feature selection for intrusion detection system using ant colony optimization," *IJ Netw. Secur.*, vol. 18, no. 3, pp. 420–432, 2016.

[38] Z. W. Geem, "Optimal cost design of water distribution networks using harmony search," *Eng. Optim.*, vol. 38, no. 3, pp. 259–277, 2006. doi: 10.1080/03052150500467430.

[39] W. M. Makki, M. M. Siraj, and N. M. Ibrahim, "A harmony search-based feature selection technique for cloud intrusion detection," in *Int. Conf. Reliable Inf. Commun. Technol.*, Springer, 2019, pp. 779–788.

[40] R. Iqbal, F. Doctor, B. More, S. Mahmud, and U. Yousuf, "Big data analytics: Computational intelligence techniques and application areas," *Technol. Forecast. Soc. Change*, vol. 153, pp. 119253, 2020. doi: 10.1016/j.techfore.2018.03.024.

[41] A. Patel, M. Taghavi, K. Bakhtiyari, and J. C. Júnior, "An intrusion detection and prevention system in cloud computing: A systematic review," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 25–41, 2013. doi: 10.1016/j.jnca.2012.08.007.

[42] V. Vepnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.

[43] S. Revathi and A. Malathi, "A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection," *Int. J. Eng. Res. Technol. (IJERT)*, vol. 2, no. 12, pp. 1848–1853, 2013.

[44] O. Osanaiye, H. Cai, K. K. R. Choo, A. Dehghantanha, Z. Xu and M. Dlodlo, "Ensemble-based multi-filter feature selection method for DDoS detection in cloud computing," *EURASIP J. Wirel. Commun. Netw.*, vol. 2016, no. 1, pp. 1–10, 2016. doi: 10.1186/s13638-016-0623-3.

[45] A. Aldribi, I. Traore, P. G. Quinan, and O. Nwamuo, "Documentation for the isot cloud intrusion detection benchmark dataset (ISOT-CID)," University of Victoria, 2020.

[46] A. Aldribi, I. Traoré, B. Moa, and O. Nwamuo, "Hypervisor-based cloud intrusion detection through online multivariate statistical change tracking," *Comput. Secur.*, vol. 88, pp. 101646, 2020. doi: 10.1016/j.cose.2019.101646.