Novel Collaborative Intrusion Detection for Enhancing Cloud Security

Widad Elbakri[1], Maheyzah Md. Siraj[2]*, Bander Ali Saleh Al-rimy[3],
Sultan Ahmed Almalki[4]*, Tami Alghamdi[5], Azan Hamad Alkhorem[6], Frederick T. Sheldon[7]
Faculty of Computing, Universiti Teknologi Malaysia, Johor Bahru, 81310, Malaysia[1,2]
School of Computing, University of Portsmouth, Portsmouth PO1 3HE, UK[3]
Computer Department-Applied College, Najran University, Najran 66462, Kingdom of Saudi Arabia[4]
Computer Science Department-Faculty of Computing and Information,
Al-Baha University, Al-Baha, 65779, Kingdom of Saudi Arabia[5]
Department of Computer Engineering-College of Computer Science and Information Technology,
Majmaah University, Al-Majmaah 11952, Kingdom of Saudi Arabia[6]
Department of Computer Science, University of Idaho, Moscow, ID 83844, USA[7]

***Abstract*—Intrusion Detection Models (IDM) often suffer from poor accuracy, especially when facing coordinated attacks such as Distributed Denial of Service (DDoS). One significant limitation of existing IDM solutions is the lack of an effective technique to determine the optimal period for sharing attack information among nodes in a distributed IDM environment. This article proposes a novel collaborative IDM model that addresses this issue by leveraging the Pruned Exact Linear Time (PELT) change point detection algorithm. The PELT algorithm dynamically determines the appropriate intervals for disseminating attack information to nodes within the collaborative IDM framework. Additionally, to enhance detection accuracy, the proposed model integrates a Gradient Boosting Machine with a Support Vector Machine (GBM-SVM) for collaborative detection of malicious activities. The proposed model was implemented in Apache Spark using the NSL-KDD benchmark intrusion detection dataset. Experimental results demonstrate that this collaborative approach significantly improves detection accuracy and responsiveness to coordinated attacks, providing a robust solution for enhancing cloud security.**

Keywords*—Cloud security; intrusion detection; collaborative model; feature selection; anomaly detection; Pruned Exact Linear Time (PELT); gradient boosting machine; support vector machine; NSL-KDD; DDoS

I. Introduction

A. Overview of Cloud Computing and Security Challenges

Cloud computing, a transformative paradigm in IT service delivery that emerged around 2010, provides on-demand access to resources such as computing power and storage via the internet [1]. This model enables organizations to avoid substantial investments in hardware and software, paying only for what they use. Despite its significant cost advantages and operational flexibility, cloud computing introduces new and complex security challenges [2]. Among these, Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) attacks stand out as critical threats that can severely compromise cloud infrastructure [3]. Ensuring robust and adaptive security mechanisms is essential to fostering trust and promoting widespread adoption of cloud technologies.

B. Limitations of Current Intrusion Detection Models (IDM)

Intrusion Detection Models (IDM) are a cornerstone of cloud security, monitoring network events to identify and respond to potential breaches. IDM can be broadly categorized as signature-based, anomaly-based, or hybrid [4]. Signature-based Models excel at detecting known attack patterns but struggle with novel or zero-day attacks [5]. On the other hand, anomaly-based Models can identify previously unseen threats but often suffer from high false alarm rates, undermining their effectiveness [6]. Hybrid approaches aim to combine the strengths of these two methods but usually inherit their limitations, leading to suboptimal performance.

Coordinated attacks, such as DDoS attacks, exacerbate these challenges. These attacks leverage multiple compromised devices to flood targeted systems with overwhelming traffic, often evading detection by isolated IDM monitoring [7]. For example, Smurf attacks exploit spoofed IP addresses to generate a flood of Internet Control Message Protocol (ICMP) replies, overwhelming target systems [5]. These challenges highlight the need for more effective, scalable, and adaptive intrusion detection solutions.

C. Research Problem and Justification

Existing IDM solutions face critical limitations when addressing cloud environments' dynamic and distributed nature. Signature-based approaches are ineffective against zero-day attacks, while anomaly-based methods often generate excessive false positives, wasting computational resources. Furthermore, traditional IDM struggles to detect coordinated attacks, such as DDoS, due to their distributed nature and lack of collaboration among monitoring systems [8], [9], [10], [11]. These gaps necessitate developing a new, collaborative approach capable of adapting to the unique security challenges of cloud environments.

D. Proposed Solution

This paper presents a novel collaborative intrusion detection model for cloud computing environments. The proposed model addresses the limitations of existing solutions by incorporating:

*Corresponding authors.

1) Advanced feature selection techniques and change point detection using the Pruned Exact Linear Time (PELT) algorithm.
2) Collaborative classifier training and update mechanisms to enhance detection accuracy.
3) Distributed attack detection and IP traffic monitoring.
4) Aggregation and feedback loops are used to refine the detection process continuously.

This model leverages a network of specialised units to offer a robust, scalable, and adaptive solution to cloud security challenges. Its collaborative nature enables the effective detection of coordinated attacks, such as DDoS, while reducing false positives and resource wastage.

E. Organization of the Paper

The remainder of this paper is organized as follows:

Section II: A comprehensive review of related work, establishing the context for the proposed research.

Section III: A detailed presentation of the proposed collaborative intrusion detection model, including its architecture and functionalities.

Section IV: Description of the datasets and experimental setup used to evaluate the model.

Section V: Results and analysis, highlighting the model's effectiveness in addressing the identified challenges.

Section VI: Conclusion and future research directions, summarizing this work's contributions and potential extensions.

By addressing the gaps above, this research aims to enhance the security and resilience of cloud computing environments, fostering trust and enabling broader adoption of this transformative technology.

II. Related Works

Within the IDM context, collaboration refers to the cooperation and communication among multiple IDM nodes or agents across different sub-networks and/or hosts. These nodes share information to detect anomalies such as coordinated attacks or Distributed Denial of Service (DDoS) attacks. A collaborative IDM has the potential to detect attacks dispersed over several hosts or networks by aggregating evidence across these sub-networks. To address the issue of coordinated attacks like DDoS, existing work in cloud IDM can be categorized into signature-based, anomaly detection, and hybrid techniques.

Several researcher teams have employed signature-based techniques for collaborative cloud IDM, such as [12] and[13]. In their approach, each region in the collaborative cloud has an IDM deployed, which interacts with others by sharing alert information aimed to mitigate the impact of DDoS attacks. For instance, the framework implemented by these aforementioned researchers uses Snort-based IDM with three plug-in modules: block, communication, and cooperation. Detection agents collate and correlate alerts to assess their accuracy through a majority vote model to enhance faulty local assessments. Once an alert is accepted, a new blocking rule is added to the block table. However, this approach can only detect known attacks due to its reliance on signature-based methods. Similarly, [14] proposed a multi-threaded distributed cloud IDM for detecting DDoS attacks, comprising modules for capture and queue, analysis, and reporting. Their experimental tests in a .NET simulator demonstrated the model's ability to identify and drop bad packets. Still, it remains limited to known attack signatures and is ineffective against zero-day attacks.

Anomaly detection approaches have also been explored by researchers such as [9], whom have proposed a distributed IDM for cloud computing using a data mining approach. In their technique, network traffic is collected from edge routers and forwarded to anomaly detection devices using a Naïve Bayes classifier, with further classification by a Random Forest classifier at a central server. The author in [15] proposed a statistical and distributed network packet filtering model against DDoS attacks involving a coordinator that distributes detection tasks among various virtual machines. The author in [16] utilized Neural Networks and the Bat algorithm in their distributed IDM, while [17] employed the artificial Bee Colony algorithm and neural networks. The author in [18] developed an egress detection model using Principal Component Analysis (PCA) to protect cloud environments from DDoS attacks, with monitoring probes in each hypervisor and a hierarchical node structure for decision-making. However, the anomaly detection approach often leads to high false alarms, as [19] and [20] noted.

The hybrid approach combines anomaly detection and signature-based techniques [21], [22]. They proposed a hybrid and collaborative IDM that uses Snort for predefined attacks and a decision tree classifier with SVM for distributed attacks [23]. This research paper proposes a hybrid machine learning technique combining the Extreme Learning Machine (ELM) model with the black hole optimization algorithm for DDoS attack detection in cloud computing. However, they also inherit certain limitations, including the challenge of high false alarms and the reliance on known signatures for some detection.

In addition, existing research proposes a distributed anomaly detection system using Gaussian Mixture-based Correntropy (i.e. an adaptive neurotechnology that measures similarity, normally utilized in statistical signal processing and is based on second order moments) to identify zero-day attacks at the edge of networks [24]. While demonstrating effective performance on specific datasets, this approach lacks adaptability and collaboration, which are crucial for dynamic cloud environments. [25] employs distributed machine learning with ensemble techniques and concept drift handling for intrusion detection. While achieving high accuracy, the approach needs to have the collaborative and adaptive capabilities to strengthen a Cloud Anomaly Intrusion Detection Model (CAIDM), to strengthen their effectiveness in dynamic and distributed environments. In contrast, our proposed CAIDM addresses these limitations and demonstrates its effectiveness by incorporating adaptive and collaborative features.

To address these limitations, this paper proposes a collaborative intrusion detection model for cloud computing using the Pruned Exact Linear Time (PELT) change point detection algorithm. This approach aims to optimize the timing for exchanging attack information among nodes in the collaborative IDM and, as shown below, enhancing the model's effectiveness and reducing false alarms.

III. Proposed Enhanced Collaboration and Adaptive Cloud Anomaly Intrusion Detection Model

The Enhancing Collaboration and Adaptive Cloud Anomaly Intrusion Detection Model (EC-A-CAIDM) orchestrates a multifaceted defense against cyber threats. This cooperative model comprises seven specialized units: i) feature selection using hybrid Harmony Search Optimization and a Symmetrical Uncertainty Filter (HSO-SUF)-based feature selection for selecting the most relevant features in the dataset, ii) change point detection utilizing Pruned Exact Linear Time (PELT), iii) collaborative classifier training and update, iv) distributed attack detection, v) IP address traffic monitoring, vi) aggregation, and vii) feedback. These units operate in a synchronized three-phase choreography: training, testing, and retraining/updating.

The core of the cooperative training phase lies within the collaborative classifier training and update unit. It orchestrates a distributed training process employing the GBM-SVM classifier within the Apache Spark framework [Gradient Boosting Machine (GBM), Support Vector Machine (SVM)]. Worker nodes individually refine their Local Normal Reference Models (LNRMs) on designated data partitions [i.e. Resilient Distributed Datasets (RDDs) as identified commonly] within the distributed dataset, akin to independent learning modules. These localized models capture patterns of normalcy specific to each node's domain. The head node, acting as a central coordinator and responsible for aggregating and analyzing the LNRMs, harmonizes these LNRMs into a unified Global Normal Reference Model (GNRM), reflecting the collective picture of normalcy across the network. The testing or detection phase translates this learned normalcy into real-time vigilance. Change point detection algorithms meticulously monitor data streams for abrupt shifts, while the distributed attack detection unit leverages the LNRMs to identify local anomalies. Exceeding a predefined threshold triggers an alert, prompting the unit to forward its LNRM to the head node for further analysis. However, the material sentinel of this phase is the IP address traffic monitoring unit acting as a watchful sentinel, tracking destination IP volume. The IP volume provides valuable insight about deviations suggestive of potential DDoS attacks, ensuring a model's comprehensive coverage by our proposed model (EC-A-CAIDM).

The final phase, retraining and updating, ensures the model continuously evolves. The aggregation unit seamlessly integrates LNRMs from all detection units into the GNRM, keeping the unit in tune with the dynamic network landscape. However, the real star of this phase is the feedback unit. It plays a crucial role toward extracting insights from detected intrusions and anomalies and feeding them into the collaborative training phase. This continuous feedback loop is essential to the model's success, enhancing future detection accuracy and ensuring it maintains its' edge (i.e. dominance) in the ever-changing cybersecurity landscape, where normally, the attackers maintain an edge that utilizes intelligent maneuverings and diversion.

Fig. 1 and 2 provide an in-depth understanding of the intricate design of the EC-A-CAIDM, while Algorithm 1 presents the pseudocode of the proposed EC-A-CAIDM. Together, Fig. 1, Fig. 2 illustrate the operational flow, showcasing novel functionalities with dashed lines. A detailed exploration of each unit and its contribution to the overall security framework is presented in the subsequent sections, offering a comprehensive insight into the model's orchestration.

IV. The EC-A-CAIDM Training Phase

The training phase of the Enhancing and Adaptive Collaboration Cloud Intrusion Detection Model (EC-A-CAIDM) lays the crucial groundwork for its exceptional effectiveness and precision. Three vital units orchestrate this phase: feature selection, collaborative classifier training, and IP address traffic monitoring. Each unit plays a distinct yet pivotal role in shaping EC-A-CAIDM's capabilities, forming the foundation upon which collaborative detection thrives.

A. Feature Selection Unit

The proposed Enhanced Collaborative and Adaptive Cloud Anomaly Intrusion Detection Model (EC-A-CAIDM) incorporates a crucial feature selection unit utilizing a hybrid approach combining Harmony Search Optimization (HSO) and Symmetrical Uncertainty Filter (SUF). *Feature selection (FS)* is essential in machine learning, particularly for intrusion detection models (IDMs). FS helps improve a machine learning based models' efficiency and accuracy by removing irrelevant or noisy data attributes that impede detection capabilities [26], [27].

The HSO algorithm plays a pivotal role for the steps defined in this process. HSO, introduced by [28], is a metaheuristic optimization technique inspired by musical improvisation. Like musicians whom collaborate to create harmony, HSO explores the search space to identify the optimal combination of features that best distinguish normal from abnormal network traffic in the cloud environment. This approach is particularly well-suited for feature selection as it efficiently finds good-to-excellent solutions within a complex search space, even though they likely differ from the absolute best (unlike some heuristic methods) [29].

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 favourable 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 continues until a predefined criterion is met, ensuring convergence to an optimal or near-optimal feature subset.

The HSO-SUF combination further enhances the feature selection process. While HSO efficiently explores the feature

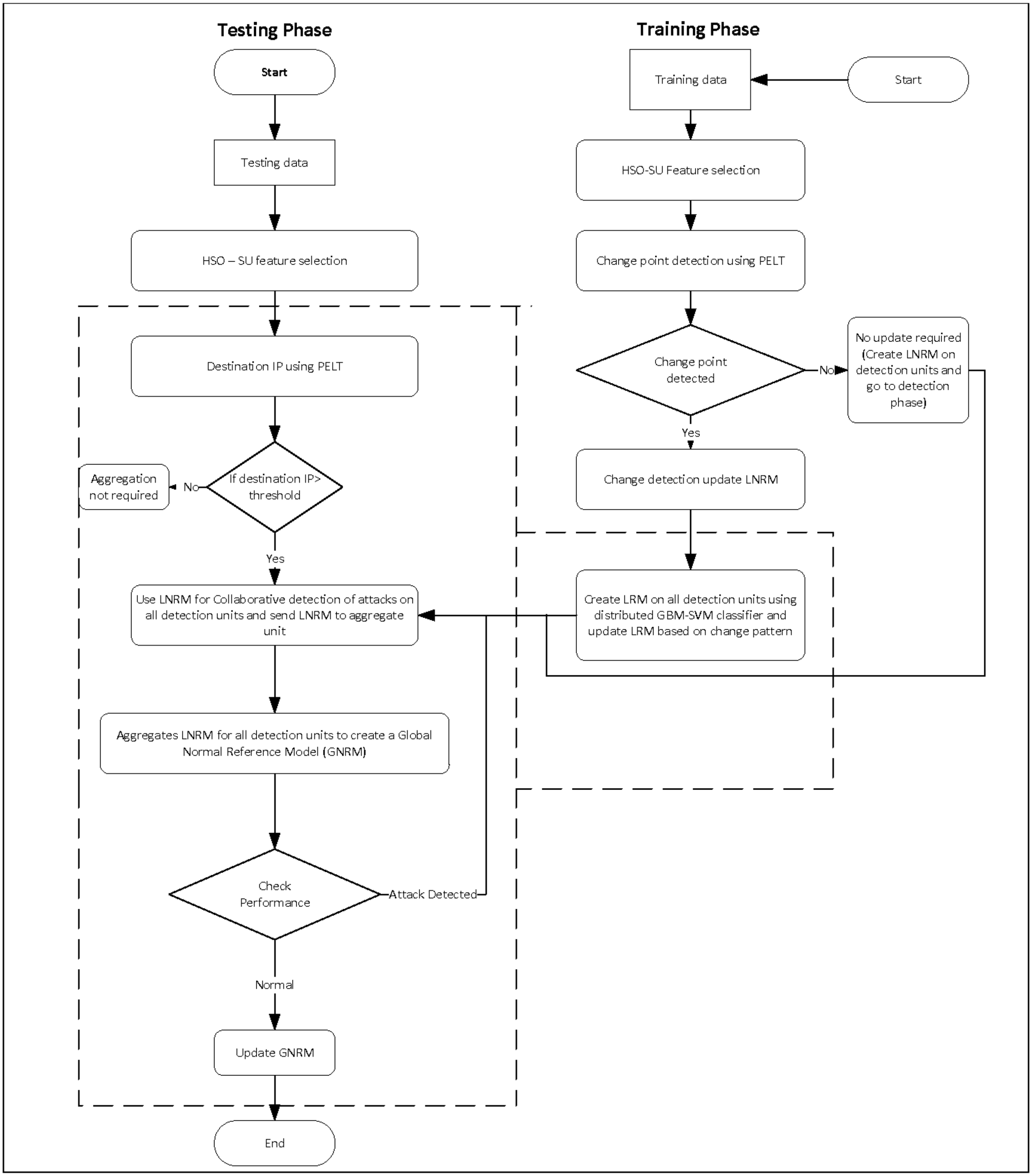
Testing Phase
Training Phase
Start
Training data
Start
Testing data
HSO-SU Feature selection
HSO – SU feature selection
Change point detection using PELT
Destination IP using PELT
Change point detected
No
No update required (Create LNRM on detection units and go to detection phase)
Yes
Change detection update LNRM
Aggregation not required
No
If destination IP> threshold
Yes
Use LNRM for Collaborative detection of attacks on all detection units and send LNRM to aggregate unit
Create LRM on all detection units using distributed GBM-SVM classifier and update LRM based on change pattern
Aggregates LNRM for all detection units to create a Global Normal Reference Model (GNRM)
Check Performance
Attack Detected
Normal
Update GNRM
End

Fig. 1. The process for the adaptive and collaborative cloud intrusion detection model.

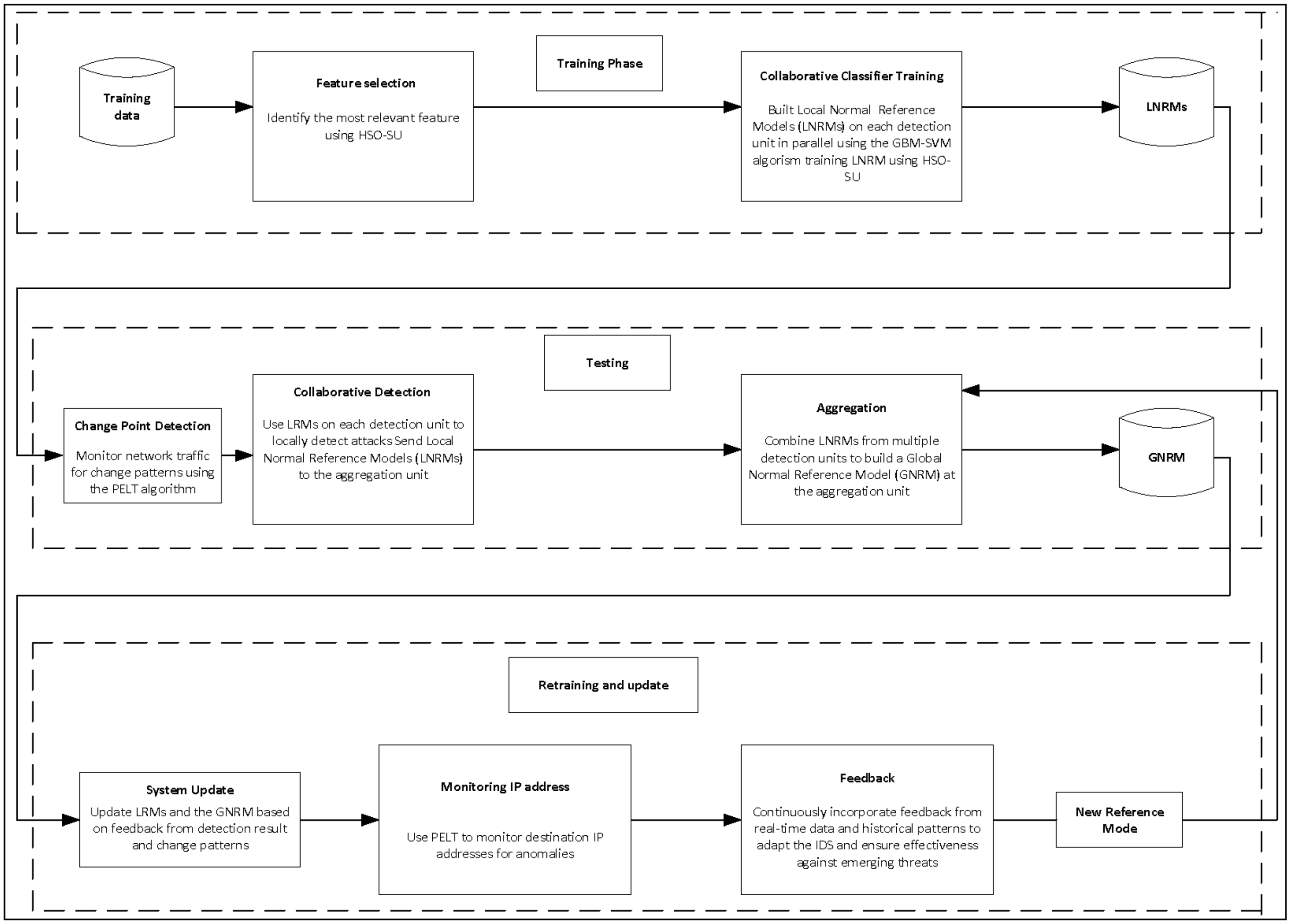
Training Phase
Training data
Feature selection
Identify the most relevant feature using HSO-SU
Collaborative Classifier Training
Built Local Normal Reference Models (LNRMs) on each detection unit in parallel using the GBM-SVM algorism training LNRM using HSO-SU
LNRMs
Testing
Change Point Detection
Monitor network traffic for change patterns using the PELT algorithm
Collaborative Detection
Use LRMs on each detection unit to locally detect attacks Send Local Normal Reference Models (LNRMs) to the aggregation unit
Aggregation
Combine LNRMs from multiple detection units to build a Global Normal Reference Model (GNRM) at the aggregation unit
GNRM
Retraining and update
System Update
Update LRMs and the GNRM based on feedback from detection result and change patterns
Monitoring IP address
Use PELT to monitor destination IP addresses for anomalies
Feedback
Continuously incorporate feedback from real-time data and historical patterns to adapt the IDS and ensure effectiveness against emerging threats
New Reference Mode

Fig. 2. Framework for an adaptive and collaborative cloud intrusion detection model with enhanced performance.

space, the SUF filter provides a mechanism to evaluate the relevance of each feature to the intrusion detection task. This two-pronged approach ensures that the selected features are diverse and demonstrably informative for anomaly detection. Our previous work [30] delves deeper into HSO for feature selection. By integrating HSO and SUF, our feature selection unit aims to optimize model performance by selecting features that significantly contribute to anomaly detection while mitigating the impact of noise and irrelevant data. This HSO-SUF combination forms a critical component of our comprehensive evaluation, demonstrating the effectiveness of HSO-SUF in addressing the unique challenges posed by cloud intrusion detection.

Therefore, the impact of Feature Selection in the process is an efficient HSO-SUF FS-filter that reduces the dataset's dimensionality, leading to a more manageable and computationally efficient intrusion detection model. This efficient approach, by focusing on relevant features, improves the model's accuracy in identifying anomalous network traffic, providing reassurance of its effectiveness.

Pheromone updating is based on the fitness function (γ') as depicted in Eq. (1). The feature subset discovered by the Harmony Search is denoted as X^i. The quality of the subset X^i and its size $|X^i|$ are measured using the evaluation metrics employed in the proposed HSO-SUF Model, namely Accuracy, Detection Rate (Precision), False Positive Rate and Sensitivity, as highlighted in Eq. (2), (3), (4) and (5). The Random Forest classifier is used to calculate metrics such as False Positives (FP), False Negatives (FN), and True Positives (TP), where FP represents the false positive rate, FN represents the false-negative rate, and TP represents the true positive rate [31].

$$\gamma' = \frac{\text{Sensitivity}\left(X^i\right) + \text{Precision}\left(X^i\right)}{|X^i|} \tag{1}$$

$$\text{Accuracy } = \frac{(TP+TN)}{(TP+TN+FP+FN)} * 100 \tag{2}$$

$$\text{Precision} = \frac{TP}{TP+FP} \tag{3}$$

$$\text{False Positive Rate} = \frac{FP}{FP+TN} \tag{4}$$

Algorithm 1 Collaborative and Adaptive Cloud Anomaly Intrusion Detection Model

```
Require: $X, Y$: Network traffic training and test data
Ensure: $Sbest$: Optimal feature set, $\tau$: Change point positions, $m$: Number of change points, $i\tau$: Interval between successive change points, $c$: Classification output normal or intrusion
(Feature Selection)
Input network traffic data $X, Y$.
Select optimal features $Sbest$.
(Collaborative Classifier Training and Update)
Input training data $X, Y$ from datastore and split data among various nodes.
Initialize model parameters.
Detection units build Local Normal Reference Model (LNRM) $m_i$ in parallel using GBM-SVM on each computing node.
Update Local Normal Reference Model using $\mu i\tau$ as the frequency of the update period.
(Change point Detection)
Detect the position of the change in data using PELT.
if change point $\tau$ is detected then
    Count the number of change points $m$.
    Determine the average interval between successive change points, $\mu i\tau$.
    For GBM-SVM, use $\mu i\tau$ as the classification model update period.
end if
(Destination-IP Traffic Monitoring)
Monitor network traffic for change patterns using the PELT algorithm.
Using the Pruned Exact Linear Time (PELT) technique, monitor the volume of destination-IP traffic from the same host.
if Destination-IP amounts of traffic from the same host ¿ Threshold then
    Send LNRM from detection units to the aggregation unit.
end if
(Collaborative Detection Phase)
for all detection units using $X, Y$ samples on the LNRM to detect attacks do
    Send Local Normal Reference Models (LNRMs) to the aggregation unit.
    Aggregate LNRM $m_i$ from all detection units by computing the average of the LNRM $M = \frac{1}{z}\sum_{z=1}^{n} m_i$.
    Return $M$.
    Classify network traffic as normal or malicious.
end for
```

$$\text{Sensitivity} = \frac{TP}{TP + FN} \tag{5}$$

B. Collaborative Classifier Training and Update Unit

This core unit orchestrates a distributed learning ballet, collaboratively training the GBM-SVM classification algorithm on the designated dataset. This collaborative approach allows the classifier model to dynamically adapt to evolving data patterns, ensuring its ongoing relevance and effectiveness. EC-A-CAIDM leverages the Python Apache Spark framework, renowned for efficiently handling distributed data processing tasks [32], [33]. To facilitate this distributed learning, the dataset is partitioned into Resilient Distributed Datasets (RDDs), distributed across a dedicated cluster comprising a head node and six worker nodes. The head node acts as the central conductor, orchestrating the collaborative efforts of the worker nodes, designated as detection units. These nodes simultaneously engage in GBM-SVM classifier training, generating their own Local Normal Reference Model (LNRM). These LNRMs capture patterns of normalcy within each node's designated data partition, providing localized insights for subsequent detection. This collaborative approach harnesses the computational power of distributed nodes, dramatically increasing efficiency and enhancing EC-A-CAIDM's overall detection capabilities.

Implementing the Collaborative Classifier training process involves partitioning datasets into Resilient Distributed Datasets (RDDs), facilitating parallelized processing by distributing data objects across clusters. This collaborative approach, characterized by seamless coordination among distributed nodes, maximizes computational efficiency, significantly enhancing the model's capability for effective intrusion detection. Moreover, the Gradient Boosting Machine (GBM) has been effectively combined with various machine learning algorithms such as Adaline, K-means, Perceptron, and Support Vector Machine (SVM) for online training [34]. In scenarios where training samples are provided sequentially, GBM processes each data sample individually, updating the model's weights accordingly [35]. The incremental nature of GBM's parameter updates for the Reference Model is a key feature that offers several advantages, including adaptability and suitability for dynamic Models that evolve over time or in scenarios where data distribution is not static [36]. Given that adaptability to the dynamic cloud environment is a crucial requirement for cloud Intrusion Detection Models (IDm) [37], this research and development effort employs GBM to achieve an adaptive Intrusion Detection Model (IDM).

Each example z in the learning task in a supervised learning context consists of a pair of instances x, y with x an arbitrary input and y an associated output. The learning process involves considering a loss function $l(\hat{y}, y)$ that quantifies the cost of prediction $\hat{y}$ errors compared to y actual outputs. This loss function plays a crucial role in the learning process, as it guides the selection of a family F of functions $f_w(x)$ with parameters w, represented as a weight vector, and the search function $f \in F$. The objective is to find the function that minimizes the average loss across all examples, as Eq. (6) describes. The training (Reference Model) performance is evaluated using empirical risk, as shown in Eq. (7). This empirical risk $E_n(f)$ measures how well the model performs on the training data. The SVM (hinge loss) is employed as the loss function [34], as illustrated in Eq. (8). GBM builds a Reference Model for classification by minimizing empirical risk $E_n\left(f_w\right)$ after each iteration (t) and updating it based on a single occurrence z_t using Eq. (9).

$$Q(x, w) = l\left(f_w(x), y\right) \tag{6}$$

$$E_n\left(f_w\right) = \frac{1}{n}\sum_{i=1}^{n} l\left(f_w\left(x_i\right), y_i\right) \tag{7}$$

$$l(\hat{y}, y) = \max(0, 1 - \hat{y}y) \tag{8}$$

$$w_{t+1} = w_t - \gamma_t \nabla_w Q(z_t, w_t) \tag{9}$$

C. IP Traffic Monitoring Unit

The focus on destination IP analysis stems from the observed surge in destination IPs from the same host during DDoS attacks [38], analyzing the mean values of destination IP addresses during normal periods. The meticulously designed training phase of EC-A-CAIDM lays the groundwork for its collaborative detection prowess. By harnessing the power of feature selection, distributed classifier training, and IP traffic monitoring, EC-A-CAIDM equips itself with the essential insights and models required for accurate and efficient anomaly detection in the intricate and diverse landscape of cloud computing.

In the PELT algorithm, detecting a single change point can be posed as a hypothesis test. The null hypothesis H_0 corresponds to no change point ($m = 0$), and the alternate hypothesis H_1 is a single change point ($m = 1$). A test statistic is constructed to decide whether a change has occurred. The likelihood ratio-based approach is used to test the hypothesis, which requires the calculation of maximum log-likelihood under both the null and alternate hypotheses. For the null hypothesis, the maximum log-likelihood is $\log \hat{P}(y_{1:n}|\theta)$, where $P(y_{1:n})$ is the probability density function associated with the data distribution, and $\hat{\theta}$ is the maximum likelihood estimate of the parameter [39]. The maximum likelihood under the alternate hypothesis is $\max_{T_1} \mathrm{ML}(T_1)$, where the maximum is taken over all possible change points, as shown in Eq. 10.

$$\lambda = 2\left[\max_{T_1} \mathrm{ML}(T_1) - \log P\left(y_{1:n} \mid \hat{\theta}\right)\right] \tag{10}$$

The test requires selecting a threshold C so that the null hypothesis λ is rejected if λ exceeds C. To identify multiple change points, the likelihood test statistic can be expanded to find the maximum of $\mathrm{ML}(T_{1:m})$ across all possible combinations of $T_{1:m}$ as shown in Eq. 11.

$$ML\left(T_{1:m}\right) = \sum_{i=1}^{m+1}\left[C\left(y_{(T_{i-1}+1):T_1}\right)\right] + Bf(m) \tag{11}$$

The cost function C represents a segment's cost, and $B_f(m)$ serves as a penalty to prevent overfitting. The negative log-likelihood is commonly used as the cost function, while Akaike's Information Criterion (AIC) and Bayesian Information Criterion (BIC) are popular choices for the penalty [39]

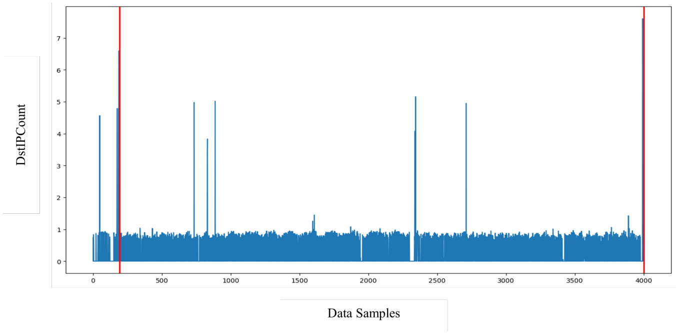

Fig. 3. Mean destination host count during normal periods in the NSL-KDD dataset.

The threshold for sending the Local Normal Reference Model (LNRM) to the aggregation unit was determined experimentally by examining the mean value of the destination-host-count (the number of network connections to the same destination host) for normal data instances using samples from the NSL-KDD dataset.

In Fig. 3, the red horizontal line indicates the mean value of the destination-host count during normal periods, which is approximately 1.0. Thus, the threshold for the normal period was set at 1.0. Thus, by integrating these advanced techniques, EC-A-CAIDM enhances its ability to detect and respond to anomalies, providing robust security for cloud computing environments.

V. The EC-A-CAIDM Testing Phase

The testing phase of EC-A-CAIDM seamlessly unites three core units i) change point detection, ii) distributed attack detection, and iii) aggregation—to create a symphony of collaborative threat identification and response. This phase rigorously evaluates the model's ability to accurately detect and address anomalies in network traffic data, with each unit playing a distinct yet complementary role in bolstering overall reliability.

A. Change Point Detection (CPD) Unit

The change point detection unit, our Model's watchful sentinel, meticulously monitors network traffic data for deviations that indicate potential intrusions or anomalies. The DAD unit vigilantly watches to identify change points within the datasets, providing crucial insights for determining an optimal frequency for updating the IDM reference model. This unit is essential to guarantee prompt and accurate threat identification by detecting subtle shifts in data patterns, enabling EC-A-CAIDM to adapt to evolving threats proactively.

B. Distributed Attack Detection (DAD) Unit

This unit acts as the heart of the EC-A-CAIDM model during the detection phase, conducting parallel intrusion detection on test data using the LNRMs generated during the distributed training phase. The DAD unit *dynamically* adjusts the transmission of LNRMs to the aggregation unit (head node) based on a carefully determined threshold established through empirical observations of traffic patterns—specifically, changes in mean traffic volume from the same host, as detailed in the traffic monitoring unit.

Apache Spark's shared variables, known as accumulators, facilitate seamless communication between detection units (agent nodes) and the aggregation unit, enabling cohesive assessment of the intrusion detection process. Additionally, EC-A-CAIDM employs a synchronized approach to optimize computational resources and response times. The transmission of results is meticulously coordinated using the mini-batch parameter of GBM-SVM, ensuring alignment with periods of heightened destination-IP volume indicative of potential threats.

C. Aggregation Unit (AU)

In this final stage of the detection phase, the aggregation unit acts as a central conductor, synthesizing individual insights into a comprehensive global perspective. The AU collects, amalgamates, and consolidates the Local Normal Reference Models (LNRMs) contributed by participating detection nodes using Eq. (6) through (9), forming a robust Global Normal Reference Model (GNRM) as shown in Eq. (12). This collaborative approach is fundamental to the core objectives of our model. By drawing upon the collective strengths of each node's unique perspective, the AU empowers EC-A-CAIDM to effectively detect anomalies and adapt to evolving threat landscapes, ultimately enhancing its overall performance and adaptability. The detection phase of EC-A-CAIDM showcases the power of collaborative intelligence needed for effective intrusion detection. EC-A-CAIDM demonstrates its ability to recognize and address risks efficiently in a dynamic and distributed cloud environment by carefully coordinating i) change point detection, ii) distributed attack detection, and iii) aggregation. This process, as elucidated by [40], is fundamental to the core objectives of our collaborative model.

$$M = \frac{1}{k}\sum_{i=1}^{k} m_i \tag{12}$$

VI. The EC-A-CAIDM Retraining and Updating Phase

The final phase of the Enhancing Adaptive and Collaborative (EC-A-CAIDM) model, retraining and updating, constitutes the driving force of its continuous evolution and adaptation to the ever-evolving threat landscape. This phase is governed by the Feedback unit, which acts as the Model's learning engine, enabling it to continuously refine its capabilities through insights extracted from real-time and historical data.

A. Feedback Unit

In an adaptive and collaborative intrusion detection model, the role of feedback mechanisms is paramount. These mechanisms gather crucial information from detected intrusions and anomalies, providing valuable fuel for model improvement. Upon detecting an event, the feedback model meticulously captures relevant data, including the nature of the threat, its specific characteristics, and the model's response. This rich data reservoir is then analyzed and processed to extract valuable insights and patterns. These extracted insights are subsequently fed into the training phase, serving as an essential input for the adaptive refinement of intrusion detection models.

This iterative feedback loop forms the cornerstone of the model's continuous learning and improvement. By incorporating experiences into its training arsenal, EC-A-CAIDM continuously updates its detection algorithms, fine-tunes decision-making processes, and expands its knowledge base on potential threats. This continuous evolution empowers the Model to consistently improve its accuracy in identifying and mitigating diverse intrusion attempts, ultimately enhancing its resilience against the ever-shifting threat landscape in dynamic cloud environments. Now, lets consider the combining of elements in the above Eq. (12) for the feedback unit, which adjusts the model continuously based on feedback from detected anomalies. The process can be summarized as shown in Eq. (13):

$$w_{t+1} = w_t - \gamma_t \nabla_w \left(\frac{1}{n}\sum_{i=1}^{n} \max(0, 1 - f_{w_t}(x_i) y_i) \right) \tag{13}$$

Here, γ_t represents the learning rate at time t, and ∇_w denotes the gradient with respect to the weight vector w. This equation ensures that the model weights are updated continuously based on the feedback from detected anomalies, thereby improving the model's performance over time.

VII. Dataset

The NSL-KDD benchmark dataset has been selected to evaluate the proposed Enhanced Adaptive and Collaborative Cloud Intrusion Detection Model (EC-A-CAIDM) due to its widespread acceptance in the research community. Despite being derived from the KDD-Cup 99 dataset and its potential limitations in representing real-world cloud intrusions, NSL-KDD offers a diverse and labelled network traffic data set that includes both normal and malicious activities. This dataset provides a standardized evaluation platform and aligns with standard research practices, reassuring the readers about the research's validity and reliability [41] and comparability.

The NSL-KDD dataset, with its realism and diversity of attack types, provides a robust tool for assessing intrusion detection models in both traditional networks and cloud computing environments. This dataset comprises 41 features with labels, including instances from KDD-Cup 99 and introduces some new attacks into the test set. The dataset includes categories such as DoS, Probe, User to Root (U2R), and Remote to Local (R2L), with detailed class distributions in the training and test sets. These comprehensive features enable a thorough and rigorous evaluation of the EC-A-CAIDM, instilling confidence in the research's methodology and results [42].

For a detailed breakdown of the dataset's composition, refer to Table I and Table II.

TABLE I. DISTRIBUTION OF INSTANCES IN THE TRAINING AND TESTING NSL-KDD DATASET

Class	Training	Testing
Normal	67,343	9711
DoS	45,927	7460
Probe	11,656	2421
R2L	995	2885
U2R	52	67
Total	**125,973**	**22,544**

VIII. RESULTS

Evaluating the Enhanced Adaptive and Collaborative Cloud Intrusion Detection Model (EC-A-CAIDM) is a practical assessment of the proposed model's real-world application. We'll explore its performance across the following key areas:

A. Feature Selection

Examines the effectiveness of the Hybrid Feature Selection (HSO-SUF) technique, which leverages Harmony Search Optimization and a Symmetrical Uncertainty Filter, for identifying the most relevant features for intrusion detection within the cloud environment. The feature selection process identified a refined set of 13 features from the original 41 features within the NSL-KDD dataset. This selection process plays a crucial role in improving the efficiency and accuracy of the intrusion detection model. In particular, these 13 features are the key to validating our improved model. The features are Network Traffic Characteristics: Service, flag, src-bytes, and dst-bytes. Connection Details: logged-in, count. Error Rates: error-count, dst-host-serror-rate, dst-host-srv-serror-rate. Traffic Patterns: same-srv-rate, diff-srv-rate, dst-host-srv-count, dst-host-same-srv-rate.

B. IP Traffic Monitoring Unit

We evaluate the performance of the PELT change point detection algorithm employed by the IP Traffic Monitoring Unit. This analysis focuses on its ability to detect anomalies in destination IP traffic patterns. The IP Traffic Monitoring Unit, with the crucial assistance of the PELT change point detection algorithm, identifies significant changes in the mean volume of traffic directed towards specific destination IPs originating from the same source. This analysis, driven by the PELT algorithm, is key in determining when to trigger the aggregation of Local Normal Reference Models (LNRMs) from various detection units. As illustrated in Fig. 4, the destination host count surpasses the pre-defined normal threshold established during the destination-IP monitoring phase. This significant threshold violation between the 1000th and 2300th instances in the data demands immediate action. Consequently, during this specific period, the detection units will send their LNRMs to the aggregation unit for further processing.

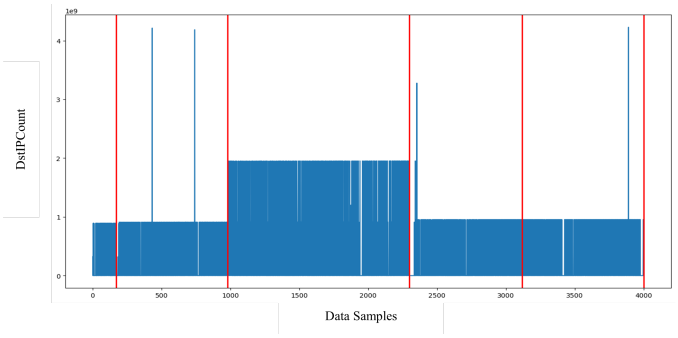

Fig. 4. Mean destination host count during Smurf DDoS attacks from NSL-KDD.

IX. DISCUSSION

The results of the proposed Enhanced Collaborative and Adaptive Cloud Anomaly Intrusion Detection Model (EC-A-CAIDM) demonstrate significant improvements in performance metrics, as shown in Table III. The EC-A-CAIDM achieved an impressive accuracy of 100%, a detection rate of 99.99%, and a false positive rate of 0.01%. This 100% accuracy is a significant achievement, indicating that the model is highly effective in identifying and mitigating such intrusions in cloud environments. It is important to note, that this accuracy was achieved on a comprehensive and diverse dataset, ensuring that the model does not over fit a specific set of data.

Comparative analysis with existing collaborative anomaly intrusion detection models (AIDMs) highlights the superior performance of EC-A-CAIDM. For instance, the Distributed Collaborative Intrusion Detection System (D-CIDS) achieved a notable accuracy of 99.6%, a detection rate of 99.7%, and a false positive rate of 0.03%. While D-CIDS shows strong performance, EC-A-CAIDM outperforms the D-CIDS across all metrics, emphasizing its enhanced detection capabilities and lower false positive rate.

Similarly, the Hybrid Detection Classifier (HDC) proposed by [8], combining KNN and SVM, achieved an accuracy of 99.85%, a detection rate of 99.78%, and a false positive rate of 0.09%. Despite the effectiveness of HDC, EC-A-CAIDM's results are superior, indicating a more precise and reliable detection mechanism.

The Distributed Anomaly Detection (DAD) system, which utilizes Gaussian Mixture-based Correntropy, demonstrated high performance with an accuracy of 99.9%, a detection rate of 99.92%, and a false positive rate of 0.11%. Although DAD is highly effective, EC-A-CAIDM still provides better accuracy and a significantly lower false positive rate, making it a more effective choice for cloud anomaly detection.

Lastly, the Distributed Anomaly Detection using the Ensemble Hybrid (DADEH) technique achieved an accuracy of 93%, a detection rate of 99%, and a false positive rate of 0.3%. The notable performance gap between DADEH and EC-A-CAIDM further highlights the latter's advancements in accuracy and false positive reduction.

The EC-A-CAIDM's novel approach to determining the optimal timing for sharing attack information among nodes in the collaborative AIDM has contributed to its enhanced

TABLE II. NSL-KDD Dataset Features

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			

performance. By effectively synchronizing attack information dissemination and employing adaptive learning techniques, the model achieves higher accuracy and detection rates and significantly reduces the occurrence of false positives. This adaptive and collaborative approach ensures that the model remains robust against evolving threats and maintains high performance in diverse cloud environments, reassuring the audience.

In conclusion, the EC-A-CAIDM sets a new benchmark and standard in cloud anomaly intrusion detection by achieving outstanding performance metrics. Compared to existing models, its superior accuracy, detection rate, and low false positive rate underscore its potential as a highly effective solution for ensuring the security of cloud-based systems. The efficiency of EC-A-CAIDM is sure to impress all stakeholders.

The data in Table III and Fig. 5, 6 and 7 unequivocally demonstrate that EC-A-CAIDM outperforms these existing models regarding accuracy, detection rate, and false positive rate. Additionally, EC-A-CAIDM has a lower detection time in seconds compared to CAIDM. The superior performance of the EC-A-CAIDM can be attributed to the effective strategy in determining the optimal timing for sharing attack information among nodes in the collaborative IDM. One of the critical challenges in collaborative IDMs is deciding when to share attack information among detection units to minimize the rate of false alarms that can result from inappropriate timing. EC-A-CAIDM successfully addresses this challenge, leading to the aforementioned enhanced performance.

X. Conclusion

This paper introduced the Enhanced Collaborative and Adaptive Cloud Anomaly Intrusion Detection Model (EC-A-CAIDM), a pioneering approach crafted to combat the escalating threat of sophisticated attacks in cloud environments. EC-A-CAIDM operates on a distributed architecture with seven specialized units, including feature selection, collaborative classification, IP traffic monitoring, change point detection, distributed attack detection, aggregation units, and a feedback mechanism for continuous learning. The core strength of EC-A-CAIDM lies in its strategic sharing of attack information among detection units, guided by the PELT change point detection algorithm, which effectively mitigates the challenge of false alarms prevalent in collaborative intrusion detection models. Its comprehensive evaluation using the NSL-KDD dataset demonstrates superior accuracy, detection rate, and very low false positive rate compared to existing models, instilling confidence in its effectiveness. However, the model's reliance on predefined feature sets may limit its adaptability to zero-day attacks and advanced persistent threats (APTs), and its computational complexity could pose challenges in highly dynamic cloud environments. Future research should address these limitations by exploring adaptive feature engineering techniques, lightweight architectures for real-time processing, and adversarial learning methods to enhance resilience. Advancing these areas can further improve EC-A-CAIDM's robustness and scalability, contributing to the evolving field of cloud security and intrusion detection.

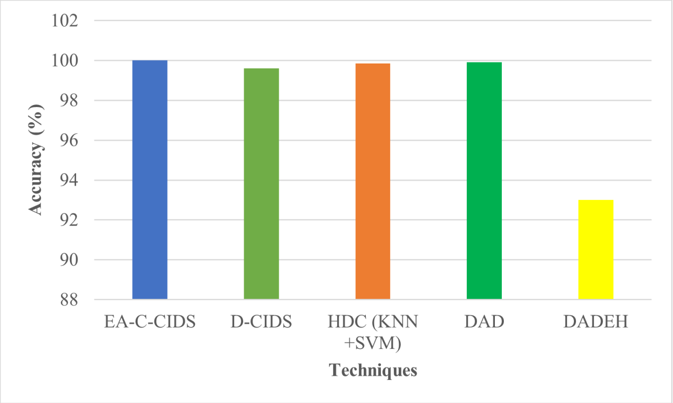

Fig. 5. Comparison of accuracy between the proposed EC-A-CAIDM vs. D-CIDM and HDC (KNN + SVM)). Example legend text.

TABLE III. COMPARISON ANALYSIS OF EC-A-CAIDM ON THE NSL-KDD DATASET WITH PREVIOUS WORK

Metrics	EA-C-CIDS	D-CIDS[43]	HDC (KNN + SVM)[8]	DAD[24]	DADEH[25]
Accuracy (%)	100	99.6	99.85	99.9	93
Detection Rate (%)	99.99	99.7	99.78	99.92	99
False Positive Rate (%)	0.01	0.03	0.09	0.11	0.3

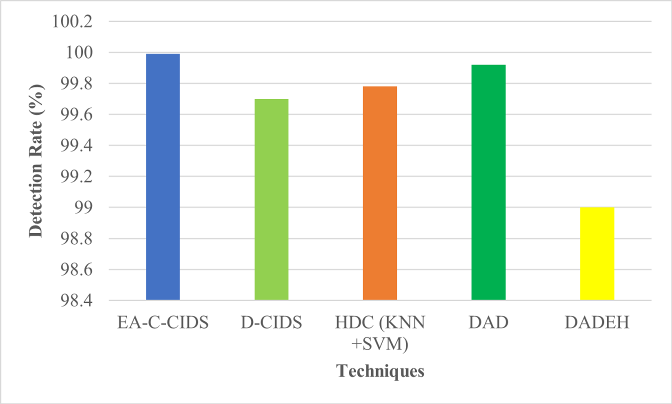

Fig. 6. Comparison of detection rate between the proposed EC-A-CAIDM vs. D-CIDM and HDC (KNN + SVM). Example legend text.

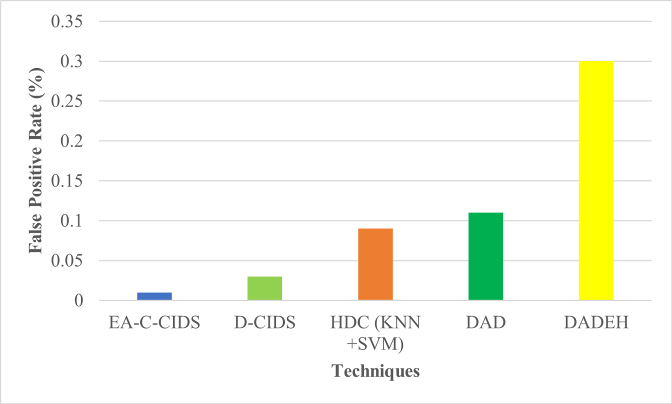

Fig. 7. Comparison of false positive rate between the proposed EC-A-CAIDM vs. D-CIDM and HDC (KNN + SVM)). Example legend text.

REFERENCES

[1] P. Mell, T. Grance *et al.*, "The nist definition of cloud computing," 2011.

[2] U. A. Butt, R. Amin, M. Mehmood, H. Aldabbas, M. T. Alharbi, and N. Albaqami, "Cloud security threats and solutions: A survey," *Wireless Personal Communications*, vol. 128, no. 1, pp. 387–413, 2023.

[3] P. Mishra, E. S. Pilli, V. Varadharajan, and U. Tupakula, "Intrusion detection techniques in cloud environment: A survey," *Journal of Network and Computer Applications*, vol. 77, pp. 18–47, 2017.

[4] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. De Alvarenga, "A survey of intrusion detection in internet of things," *Journal of Network and Computer Applications*, vol. 84, pp. 25–37, 2017.

[5] Y. Otoum and A. Nayak, "As-ids: Anomaly and signature based ids for the internet of things," *Journal of Network and Systems Management*, vol. 29, no. 3, p. 23, 2021.

[6] W. Li, W. Meng, and L. F. Kwok, "Surveying trust-based collaborative intrusion detection: state-of-the-art, challenges and future directions," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 1, pp. 280–305, 2021.

[7] C. V. Zhou, C. Leckie, and S. Karunasekera, "A survey of coordinated attacks and collaborative intrusion detection," *computers & security*, vol. 29, no. 1, pp. 124–140, 2010.

[8] K. Samunnisa, G. S. V. Kumar, and K. Madhavi, "Intrusion detection system in distributed cloud computing: Hybrid clustering and classification methods," *Measurement: Sensors*, vol. 25, p. 100612, 2023.

[9] M. Idhammad, K. Afdel, and M. Belouch, "Distributed intrusion detection system for cloud environments based on data mining techniques," *Procedia Computer Science*, vol. 127, pp. 35–41, 2018.

[10] O. Achbarou, M. A. El Kiram, O. Bourkoukou, and S. Elbouanani, "A new distributed intrusion detection system based on multi-agent system for cloud environment," *International Journal of Communication Networks and Information Security*, vol. 10, no. 3, p. 526, 2018.

[11] O. Osanaiye, K.-K. R. Choo, and M. Dlodlo, "Distributed denial of service (ddos) resilience in cloud: Review and conceptual cloud ddos mitigation framework," *Journal of Network and Computer Applications*, vol. 67, pp. 147–165, 2016.

[12] D. Singh, D. Patel, B. Borisaniya, and C. Modi, "Collaborative ids framework for cloud," *International Journal of Network Security*, vol. 2013, 2013.

[13] Y. Wang, W. Meng, W. Li, J. Li, W.-X. Liu, and Y. Xiang, "A fog-based privacy-preserving approach for distributed signature-based intrusion detection," *Journal of Parallel and Distributed Computing*, vol. 122, pp. 26–35, 2018.

[14] I. Gul and M. Hussain, "Distributed cloud intrusion detection model," *International Journal of Advanced Science and Technology*, vol. 34, no. 38, p. 135, 2011.

[15] V. C. Pandey, S. K. Peddoju, and P. S. Deshpande, "A statistical and distributed packet filter against ddos attacks in cloud environment," *Sādhanā*, vol. 43, pp. 1–9, 2018.

[16] S. Velliangiri and J. Premalatha, "Intrusion detection of distributed denial of service attack in cloud," *Cluster Computing*, vol. 22, no. Suppl 5, pp. 10 615–10 623, 2019.

[17] U. Ali, K. K. Dewangan, and D. K. Dewangan, "Distributed denial of service attack detection using ant bee colony and artificial neural network in cloud computing," in *Nature Inspired Computing: Proceedings of CSI 2015*. Springer, 2018, pp. 165–175.

[18] P. R. Kanna, K. Sindhanaiselvan, and M. Vijaymeena, "A defensive mechanism based on pca to defend denial of-service attack," *International Journal of Security and Its Applications*, vol. 11, no. 1, pp. 71–82, 2017.

[19] Z. Liu, B. Xu, B. Cheng, X. Hu, and M. Darbandi, "Intrusion detection systems in the cloud computing: A comprehensive and deep literature review," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 4, p. e6646, 2022.

[20] O. O. Olateju, S. U. Okon, U. T. I. Igwenagu, A. A. Salami, T. O. Oladoyinbo, and O. O. Olaniyi, “Combating the challenges of false positives in ai-driven anomaly detection systems and enhancing data security in the cloud,” *Asian Journal of Research in Computer Science*, vol. 17, no. 6, pp. 264–292, 2024.

[21] H.-Y. Kwon, T. Kim, and M.-K. Lee, “Advanced intrusion detection combining signature-based and behavior-based detection methods,” *Electronics*, vol. 11, no. 6, p. 867, 2022.

[22] L. K. Vashishtha, A. P. Singh, and K. Chatterjee, “Hidm: A hybrid intrusion detection model for cloud based systems,” *Wireless Personal Communications*, vol. 128, no. 4, pp. 2637–2666, 2023.

[23] G. S. Kushwah and V. Ranga, “Distributed denial of service attack detection in cloud computing using hybridextreme learning machine,” *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 29, no. 4, pp. 1852–1870, 2021.

[24] N. Moustafa, M. Keshk, K.-K. R. Choo, T. Lynar, S. Camtepe, and M. Whitty, “Dad: A distributed anomaly detection system using ensemble one-class statistical learning in edge networks,” *Future Generation Computer Systems*, vol. 118, pp. 240–251, 2021.

[25] M. Jain and G. Kaur, “Distributed anomaly detection using concept drift detection based hybrid ensemble techniques in streamed network data,” *Cluster Computing*, vol. 24, no. 3, pp. 2099–2114, 2021.

[26] N. Biyyapu, E. J. Veerapaneni, P. P. Surapaneni, S. S. Vellela, and R. Vatambeti, “Designing a modified feature aggregation model with hybrid sampling techniques for network intrusion detection,” *Cluster Computing*, pp. 1–19, 2024.

[27] R.-H. Dong, Y.-L. Shui, and Q.-Y. Zhang, “Intrusion detection model based on feature selection and random forest,” *International Journal of Network Security*, vol. 23, no. 6, pp. 985–996, 2021.

[28] Z. W. Geem, J. H. Kim, and G. V. Loganathan, “A new heuristic optimization algorithm: harmony search,” *simulation*, vol. 76, no. 2, pp. 60–68, 2001.

[29] Z. W. Geem, “Optimal cost design of water distribution networks using harmony search,” *Engineering optimization*, vol. 38, no. 03, pp. 259–277, 2006.

[30] W. M. Makki, M. M. Siraj, and N. M. Ibrahim, “A harmony search-based feature selection technique for cloud intrusion detection,” in *Emerging Trends in Intelligent Computing and Informatics: Data Science, Intelligent Information Systems and Smart Computing 4*. Springer, 2020, pp. 779–788.

[31] M. H. Ali and M. A. Mohammed, “An improved fast learning network with harmony search based on intrusion-detection system,” *Journal of Computational and Theoretical Nanoscience*, vol. 16, no. 5-6, pp. 2166–2171, 2019.

[32] M. Assefi, E. Behravesh, G. Liu, and A. P. Tafti, “Big data machine learning using apache spark mllib,” in *2017 ieee international conference on big data (big data)*. IEEE, 2017, pp. 3492–3498.

[33] X. Meng, J. Bradley, B. Yavuz, E. Sparks, S. Venkataraman, D. Liu, J. Freeman, D. Tsai, M. Amde, S. Owen *et al.*, “Mllib: Machine learning in apache spark,” *The journal of machine learning research*, vol. 17, no. 1, pp. 1235–1241, 2016.

[34] A. Natekin and A. Knoll, “Gradient boosting machines, a tutorial,” *Frontiers in neurorobotics*, vol. 7, p. 21, 2013.

[35] J. H. Friedman, “Greedy function approximation: a gradient boosting machine,” *Annals of statistics*, pp. 1189–1232, 2001.

[36] E. J. Atkinson, T. M. Therneau, L. J. Melton III, J. J. Camp, S. J. Achenbach, S. Amin, and S. Khosla, “Assessing fracture risk using gradient boosting machine (gbm) models,” *Journal of Bone and Mineral Research*, vol. 27, no. 6, pp. 1397–1404, 2012.

[37] H. Attou, M. Mohy-eddine, A. Guezzaz, S. Benkirane, M. Azrour, A. Alabdultif, and N. Almusallam, “Towards an intelligent intrusion detection system to detect malicious activities in cloud computing,” *Applied Sciences*, vol. 13, no. 17, p. 9588, 2023.

[38] T. C. Chieu, A. Mohindra, A. A. Karve, and A. Segal, “Dynamic scaling of web applications in a virtualized cloud computing environment,” in *2009 IEEE International Conference on e-Business Engineering*. IEEE, 2009, pp. 281–286.

[39] R. Killick, P. Fearnhead, and I. A. Eckley, “Optimal detection of changepoints with a linear computational cost,” *Journal of the American Statistical Association*, vol. 107, no. 500, pp. 1590–1598, 2012.

[40] J. Duchi, E. Hazan, and Y. Singer, “Adaptive subgradient methods for online learning and stochastic optimization.” *Journal of machine learning research*, vol. 12, no. 7, 2011.

[41] S. Revathi and A. Malathi, “A detailed analysis on nsl-kdd dataset using various machine learning techniques for intrusion detection,” *International Journal of Engineering Research & Technology (IJERT)*, vol. 2, no. 12, pp. 1848–1853, 2013.

[42] 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 Journal on Wireless Communications and Networking*, vol. 2016, pp. 1–10, 2016.

[43] N. M. Ibrahim and A. Zainal, “A distributed intrusion detection scheme for cloud computing,” *International Journal of Distributed Systems and Technologies (IJDST)*, vol. 11, no. 1, pp. 68–82, 2020.