International Journal of Advanced Technology and Engineering Exploration, Vol 11(121)
ISSN (Print): 2394-5443 ISSN (Online): 2394-7454
http://dx.doi.org/10.19101/IJATEE.2024.111100395

Research Article

Adaptive intrusion detection framework for enhanced cloud security in fog and edge computing environments

K. Vani[1]* and S. P. Swornambiga[2]
Department of Computing, Coimbatore Institute of Technology, Coimbatore, Tamil Nadu, 641014, India[1]
Department of Computer Applications, CMS College of Science and Commerce, Tamil Nadu, 641049, India[2]

Received: 28-March-2024; Revised: 13-December-2024; Accepted: 17-December-2024

Abstract
With increased cloud usage and complex cyber threats, developing robust, dynamic intrusion detection systems for cloud environments is crucial. Traditional centralized solutions face latency and scalability issues. Recent frameworks, though significant, often struggle with emerging threats and inconsistent performance. This study aims to develop an adaptive intrusion detection framework to enhance cloud security by addressing the limitations of centralized detection systems and improving the detection of known and unknown threats. To remove the reliance on centralized resources, the architecture uses an intrusion detection module at fog and edge nodes to perform analysis locally, facilitating rapid identification and response. The proposed adaptive framework can be deployed in the edge/fog layer. It has three primary phases: data collection and preprocessing, training, and testing. The framework adopts a multi-layered defence strategy, employing a rule-based ensemble model for detecting known attacks, a signature-based detection method, an isolation forest (IF) model for anomaly-based detection of unknown attacks, and an ensemble learning model to make final decisions on the network traffic data. The performance analysis was conducted on the university of new south Wales network-based 2015 (UNSW-NB15) dataset with 82,332 instances and the national security laboratory - knowledge discovery and data mining (NSL-KDD) dataset with 25,192 instances. With the UNSW-NB15 dataset, the model achieved 92.68% accuracy, an 88.03% attack detection rate (ADR), and a low false alarm rate (FAR) of 1.53%. For the NSL-KDD dataset, the model offered an improved accuracy of 99.51%, along with a mean F-measure of 91.48%. It also achieved an ADR of 99.23% and a FAR of 0.25%. The proposed adaptive model outperforms conventional models in detecting security breaches, enhancing cloud security with scalability, agility, and resilience in fog and edge computing environments. However, the framework requires continuous development to improve its efficiency, and evaluate performance in real-time with large network traffic data.

Keywords
Intrusion detection system, Cloud security, Fog computing, Edge computing, Anomaly detection, Ensemble learning.

1.Introduction
Recently, the rapid growth of smart devices has led to the emergence of the internet of things (IoT) [1]. The cloud model provides users with powerful networked and remote computing resources, eliminating the need for planning, purchasing, and maintaining these resources [2]. Cloud computing faces challenges in managing IoT data, meeting performance requirements, and maintaining geo-distribution, mobility support, minimal latency, and location awareness for IoT applications [3]. Edge computing (EC) and fog computing (FC) are needed to integrate IoT frameworks and cloud systems to solve IoT deployment issues [4, 5].

Those nodes that can do computations are known as mobile EC hosts, fog nodes, and FC hosts and have the potential to provide lower-latency services. FC operates between edge devices and centralized cloud infrastructure, while EC processes data closer to its source [6, 7]. However, these models face several potential threats, such as man in the middle (MIM), denial of service (DoS), privacy leakage, service manipulation, and rogue gateways, which can be reprogrammed to disseminate false information or crowd-sourced services [8–10]. Thus, security is a significant challenge in cloud, edge, and fog paradigms, with intrusions posing a common security threat that can create vulnerabilities in the cloud environment [11, 12].

*Author for correspondence

An intrusion detection system (IDS) is utilized to detect cyberattacks [13]. Generally, there are two methods by which the IDS can identify malicious users in FC and EC environments [5, 14]. First, is signature-based intrusion detection, which identifies known attacks by comparing the incoming network traffic with a predefined set of rules that are generated based on known attack patterns. The second is anomaly-based detection, which examines the behaviour of the incoming network traffic and classifies it as normal or abnormal attack traffic. Intrusion detection solutions for FC and EC environments have been the subject of numerous research studies in recent years [10, 15]. Traditional intrusion detection solutions are centralized, causing latency and scalability concerns in large cloud environments with huge data volumes and different network architectures [16]. Similar to any other field that has embraced supervised and unsupervised learning, IDS has also employed machine learning (ML) algorithms [17]. Some of the common classifiers used in intrusion detection are adaptive boosting (AB), extreme gradient boosting or XGBoost (XGB), support vector machine (SVM) or sequential minimal optimization (SMO), classification and regression trees (CART), iterative dichotomiser (ID3), multinominal classifier (MNC), naïve Bayes (NB), decision tree (DT), random forest (RF), and deep learning (DL) classifiers such as neural networks (NN), deep belief network (DBN), convolutional neural network (CNN), long short-term memory (LSTM), multilayer perceptron (MLP), and more. However, they may struggle with real-time processing and high computational overhead.

Many recent studies have proposed state-of-the-art detection models. However, several IDS approaches, including those using DL, ensemble, or hybrid methods, suffer from high computational complexity and resource demands, which can impact real-time processing and scalability [18–20]. Models tested on specific datasets often struggle with generalization across various real-world scenarios, impacting their effectiveness across diverse network environments and attack patterns [21]. Many ML-based IDS approaches generate high false positive (FP) rates, reducing their practical applicability and reliability [22–24]. Some systems face challenges with scalability, adaptability, and performance, particularly in dynamic and resource-constrained environments [25–27]. Few other studies have a limited focus on effective feature selection, which has led to poor performance in detecting significant threats and is influenced by the quality and representativeness of the training data used [28–30]. Moreover, previous studies have explored improved detection approaches to enhance model performance; however, the security and integrity of cloud-based systems are at risk due to their inability to respond to emerging cyber threats [14, 31, 32]. The motivation for this study stems from the limitations of current IDS models in FC and EC environments, such as outdated datasets, high FP, and significant computational overhead. These challenges hinder the effective detection of contemporary threats and real-time processing, necessitating advancements in IDS frameworks for improved performance and security.

The primary objective of this research study is to design an adaptive IDS for cloud computing environments. The framework is scalable and can function effectively in decentralized FC and EC environments where centralized resources may be limited or unavailable. The proposed model applies the relief algorithm for feature selection and cluster-based representative sampling (CRS) for splitting the data. The proposed adaptive framework employs a multi-layered defense strategy by integrating a rule-based ensemble (RBE) model for signature-based detection to identify known attacks and an isolation forest (IF) classifier for anomaly-based detection to uncover suspicious patterns in network traffic. Ensemble learning is then used to make final decisions on incoming network traffic based on the outputs of the RBE and IF models. The empirical findings from the experimental evaluation on real-world datasets demonstrate the effectiveness of the proposed model in detecting and addressing evolving cyber threats.

Thus, this research aims to:

1. Develop an adaptive intrusion detection framework suitable for operating in FC and EC environments.
2. Evaluate the effectiveness of the individual defence layers of the proposed framework in detecting known and unknown cyber threats.
3. Assess the performance of the proposed adaptive IDS model in terms of average accuracy, attack accuracy, attack detection rate (ADR), false alarm rate (FAR), and other relevant metrics.
4. Compare the performance of the proposed adaptive IDS with that of traditional and state-of-the-art models.
5. Evaluate the security and complexity of the proposed adaptive IDS model.
6. Identify potential opportunities for improving the proposed framework based on empirical results.

The contributions of this research study to both theory and practice are as follows:

- Introduces a novel framework for FC and EC environments, integrating adaptive mechanisms to address unique challenges and providing practical solutions for securing these systems.
- Evaluates the effectiveness of defence layers in detecting cyber threats, offering theoretical and practical insights for practitioners to select and implement effective security measures.
- Assesses the IDS model's performance using comprehensive metrics, providing practical benchmarks for real-world comparison and understanding its strengths and limitations.
- Compares the proposed adaptive IDS model with traditional and state-of-the-art models, enhancing theoretical knowledge and guiding practitioners in adopting more effective intrusion detection technologies.
- Analyses the security features and complexity of the proposed model, providing theoretical insights and practical assessments to help practitioners understand and manage its implementation and maintenance.
- Provides empirical results that support theoretical advancements in IDS, offering actionable recommendations for future improvements and informed decision-making.

The paper is organized as follows: Section 2 reviews related work in the proposed research field. Section 3 elaborates on the proposed methodology for an adaptive IDS, detailing the framework and algorithms. Section 4 covers the experimental setup, datasets, evaluation metrics, results, and security analysis. Section 5 discusses the results, limitations, and implications of the study. Finally, section 6 concludes the work.

2.Literature review

An extensive literature analysis in the field of IDS in the cloud environment indicates the significance of the field. Several studies have been carried out by researchers on IDS for cloud security and IoT networks [33]. Earlier intrusion detection methods used statistical analysis for effective performance. Later, with the advent of ML, several learning techniques were employed to improve predictions [34]. Panda et al. (2010) [35] utilized discriminative multinomial NB (MNB) and filtering analysis to develop a network IDS using the knowledge discovery and data mining (KDD) dataset, created by the National Security Laboratory (NSL) through collaborative efforts involving researchers from the University of New Brunswick, Canada. This model, based on two class classifications and 10-fold cross-validation, shows high accuracy and a low FP rate compared to existing methods. The study was limited by its reliance on a single model and its inability to adapt to evolving threats. Generally, rule-based classifiers are widely used to detect signature-based known attacks, whereas one-class support vector machines (OC-SVM) or IF are used for detecting anomalies that are not known [36]. Ensemble learning was also widely adopted as it effectively predicted the results with multiple classifiers [23, 37]. Khraisat et al. (2020) [18] developed a hybrid IDS (HIDS) that integrates a C5.0 classifier and OC-SVM, achieving high detection accuracy and low FAR. The study struggles with computational overhead due to its specific dataset reliance.

Later, DL algorithms such as DBNs [38] and artificial NN [39] were also used along with ML models to improve attack detection accuracy [40]. Papamartzivanos et al. (2018) [41] introduced derivation of efficient network detection rules by optimized neural methods (DENDRON), a methodology utilizing DTs and genetic algorithms to develop new detection rules for common and rare attacks. They used heuristic methods to tackle network traffic challenges and prevent ML techniques from neglecting minority classes, revealing that DENDRON outperforms other techniques in classification metrics and rare intrusive incidents. Due to model complexity and dataset dependence, this study may overfit and limit its applicability in various scenarios. A study compared five classifiers using the University of New South Wales Network-Based 2015 dataset (UNSW-NB15) and the KDD 1999 (KDD'99) dataset, concluding that UNSW-NB15 is more complex, making it a valuable benchmark for evaluating network IDS [42]. The study's limited applicability to various real-world network environments and attack scenarios could be attributed to its dependence on particular datasets. A new IDS based on misuse has been developed to identify network intrusions, outperforming DT-based models and real-time datasets across various metrics [43]. However, the integrated rule-based system may struggle with scalability and adaptability to evolving threats in dynamic network conditions.

A cloud IDS was proposed to enhance classification accuracy by improving feature selection and weighting the ensemble model with the Crow search algorithm (CSA), combining filter and automated models with ML and DL models [25]. The

framework was compared to various state-of-the-art classifiers on network datasets, demonstrating its robustness at FP and false negative (FN) rates. Scalability and adaptability issues in various cloud environments and changing attack patterns could pose difficulties for the model. The study introduced a generalised isolation forest (GIF), a generalization of IF based on extended isolation forest (EIF), to address issues like lost information in EIF trees [28]. Detailed analysis indicated that GIF outperforms EIF in anomaly detection simulations, demonstrating comparable performance with reference databases. The model struggles with high-dimensional data and requires extensive tuning for optimal performance in various contexts. A DL-based method detects cybersecurity threats in IoT medical communication networks using MLP and principal component analysis (PCA), achieving a 96.39% accuracy rate and reduced time complexity [19]. The complexity of the model may lead to high computational overhead, impacting real-time detection in resource-constrained medical devices.

Similarly, Alzubi et al. (2024) [20] proposed a blended DL framework as an accurate attack detection system in the edge-centric IoT medical industry. The model, which combines a CNN and LSTM, accurately recognizes and defends healthcare data from intruders. The results indicated that the model effectively detects network attacks at the network edge with high accuracy and efficiency, outperforming existing approaches. However, the blended DL framework requires extensive computational resources, limiting its applicability in resource-constrained edge environments. A security framework was proposed employing a Markov model and virtual honeypot device (VHD) to identify malicious edge devices in FC environments, categorizing them into four levels, and maintaining log repositories [44]. The model, tested against real attacks in a virtual environment using OpenStack and Microsoft Azure, effectively identifies malicious devices and minimizes FAR. However, the model may struggle with scalability and adaptability to diverse malicious behaviours in rapidly changing FC environments.

Kumar et al. (2023) [22] suggested a study that used a better IDS model with several ML models to sort attacks on edge-of-things (EoT) networks. However, this enhanced IDS may face challenges with FPs and real-time processing in dynamic edge environments. The research utilized a filter-based technique and a combination-based grouping method to select optimally reduced features, resulting in superior performance of the RF classifier model. A novel secured edge computing IDS (SEC-IDS) was proposed that works in mobile EC environments [26]. This framework enhances intrusion detection accuracy and adaptability by combining signature-based and anomaly-based methods, utilizing EC resources for real-time response in a simulated mobile EC environment. This system may face challenges with scalability and adaptability to diverse mobile EC environments. Shitharth et al. (2023) [45] developed a multi-attack IDS for edge-assisted IoT, utilizing a back propagation NN for anomaly detection and attack methods. However, the algorithm may require extensive training data to accurately detect multi-stage attacks in edge-assisted IoT.

A HIDS model for edge-based industrial IoT was discussed that makes use of ML and DL techniques, covering a theoretical analysis without implementation [29]. However, the HIDS may face challenges with data quality and feature selection for optimal performance in EoT. A hybrid approach that makes use of ML models and blockchain technology was proposed by Liang et al. (2020) [46]. Though this method offers improved results, it takes more time to respond. An edge-based hybrid intrusion detection framework (EHIDF) was proposed using ML for real-time detection of intrusive traffic in the mobile EC environment [15]. The framework has three intrusion detection modules that use C4.5, NB, and Meta-AB classifiers to make it safer and more effective. The EHIDF, developed to detect new unknown attacks with a low false-positive reduction rate, has improved accuracy by up to 10.78% and reduced FAR by up to 93%. This method serves as the foundation for the proposed method. The study does not provide a comprehensive evaluation of the framework's performance under varying network conditions, attack patterns, and resource constraints.

A lightweight IDS was proposed that uses an MLP for FC environments [47], yet the model has higher computational complexity and detection time. A recurrent neural network (RNN) was proposed to predict intrusions from fog environments [48]. The authors reported that the model offered improved performance even with large datasets. However, the system may struggle with high computational demands and may not effectively handle diverse IoT network conditions. Kumar et al. (2022) [49] developed a distributed IDS for detecting distributed denial of service (DDoS) attacks on a blockchain-

enabled IoT network, comparing RF and XGB classifiers, revealing RF's superiority in multi-attack detection. This system may face challenges with scalability and performance under high-volume DDoS attacks in IoT networks. The authors proposed a novel unified IDS for IoT environments, defending networks from exploits, DoS, probe, and generic attacks with a higher ADR compared to existing approaches [50]. This unified IDS may struggle with high FP rates and limited adaptability to evolving IoT threats. A study proposed an FC-based smart farming framework that uses unmanned aerial vehicles (UAVs) to collect IoT sensor data, offload it at fog sites, and reduce malicious tokens, achieving an accuracy of 99.7% [27]. The framework may face challenges with scalability and may not effectively address diverse energy-based attack scenarios.

A study introduced a dynamic IDS that utilizes a multiclass SVM for real-time responses to cyberattacks on critical information infrastructure, achieving high accuracy (97.64%) and detection rates (99.20%) [21]. However, the reliance on specific datasets may limit the generalizability of the findings across diverse environments. An adaptive IDS model was suggested that combines CNN and C5.0 classifiers, demonstrating improved detection performance [50]. A limitation is the potential computational complexity and resource demands associated with deploying such models in real-time applications. Another study employed an ensemble of rule learners and a best-first search algorithm for feature selection, enhancing detection capabilities [30]. However, the system's performance may be influenced by the quality and representativeness of the training data used. A study introduced a fusion-based anomaly detection system using a modified IF tailored for IoT environments, showing effective anomaly detection [24]. A limitation is the potential for FPs in diverse IoT scenarios, which may affect the system's reliability in practical applications.

From the literature analysis, it is clear that many of the IDS models use outdated datasets for evaluating their performance. These datasets do not include recent attacks and only include a few attack types. Furthermore, the presence of anomalies in these old datasets has received little concern. Conversely, when designing an IDS, several recent studies have considered the UNSW-NB15 dataset for analysis. However, the accuracy and ADR of these IDS models are relatively low. Few earlier research studies focused only on detecting DoS attacks and overlooked other types of attacks in the cloud and fog environments [51, 52]. Moreover, few of the recent studies focused only on signature-based or anomaly-based intrusion detection, which leads to poor overall accuracy. Some methods use DL to improve attack detection; however, these models suffer from higher computational overhead. Despite the abundance of features in network traffic data, very few methods concentrate on selecting significant features before attack detection. Even though the models use traditional feature selection methods that fail to detect significant features, this leads to incorrect attack predictions. Thus, to overcome the shortcomings found in the existing studies, this work suggests an adaptive IDS model for securing the edge/fog environment that includes effective feature selection, data splitting, and multilayer classification models.

3.Method

To protect the cloud environment and its data against evolving cyber threats, an adaptive IDS framework has been proposed for securing cloud computing environments. To monitor network traffic and device behaviour in real time, this type of architecture makes use of an IDS agent or component that is deployed at edge or fog nodes. Cybersecurity strategy and defence against emerging cyber threats are being revolutionized by the implementation of IDS in fog and edge environments instead of the cloud. These IDS modules ensure instant attack detection and response without depending on centralized resources by performing data analysis locally. This method not only improves cloud security but also reduces network latency and enhances overall system performance. Deploying adaptive IDS at the edge/fog environment makes them and the cloud infrastructure more resilient by identifying and handling security breaches in unpredictable situations. The overview of the proposed adaptive IDS is presented in *Figure 1*.

Generally, the architecture can be divided into three distinct layers: the device layer, the edge/fog layer, and the cloud layer.

1. **Device layer:** This layer represents the end users and their smart devices. It includes various smart devices, such as phones, tablets, IoT devices, and other end-user gadgets. These devices are connected to the cloud through the edge/fog layer, generating data that needs to be processed and potentially monitored for security threats.
2. **Edge/Fog layer:** This middle layer is crucial as it contains the edge/fog nodes that process and analyze data close to the source, enhancing real-time responsiveness. The networking hardware ensures reliable and efficient communication and

data transmission between devices and nodes. Notable components in this layer include:

- *Edge/Fog Nodes:* These are the primary units where data processing and analysis occur locally, minimizing the need for data transmission to the central cloud.
- *Computing Resources:* This includes central processing units (CPUs) and graphical processing units (GPUs) that handle computational tasks necessary for data analysis and intrusion detection.
- *Data Storage:* Local storage units provide quick access and temporary storage of data for real-time processing.
- *Networking Devices:* Routers, switches, and other networking hardware ensure reliable and efficient communication and data transmission between devices and nodes.
- *Virtualization:* This technology allows the deployment and management of software applications across different hardware resources, ensuring flexibility and scalability.
- *IDS Components:* These modules are responsible for real-time intrusion detection, employing various methods such as signature-based, anomaly-based detection, and ensemble learning techniques for behaviour analysis.

3. **Cloud layer:** This layer stores data in the cloud, acting as the central repository for long-term storage and analysis. While the edge/fog layer handles real-time processing and immediate threat detection, this layer is responsible for long-term data storage, comprehensive data analysis, and management of the IDS framework.

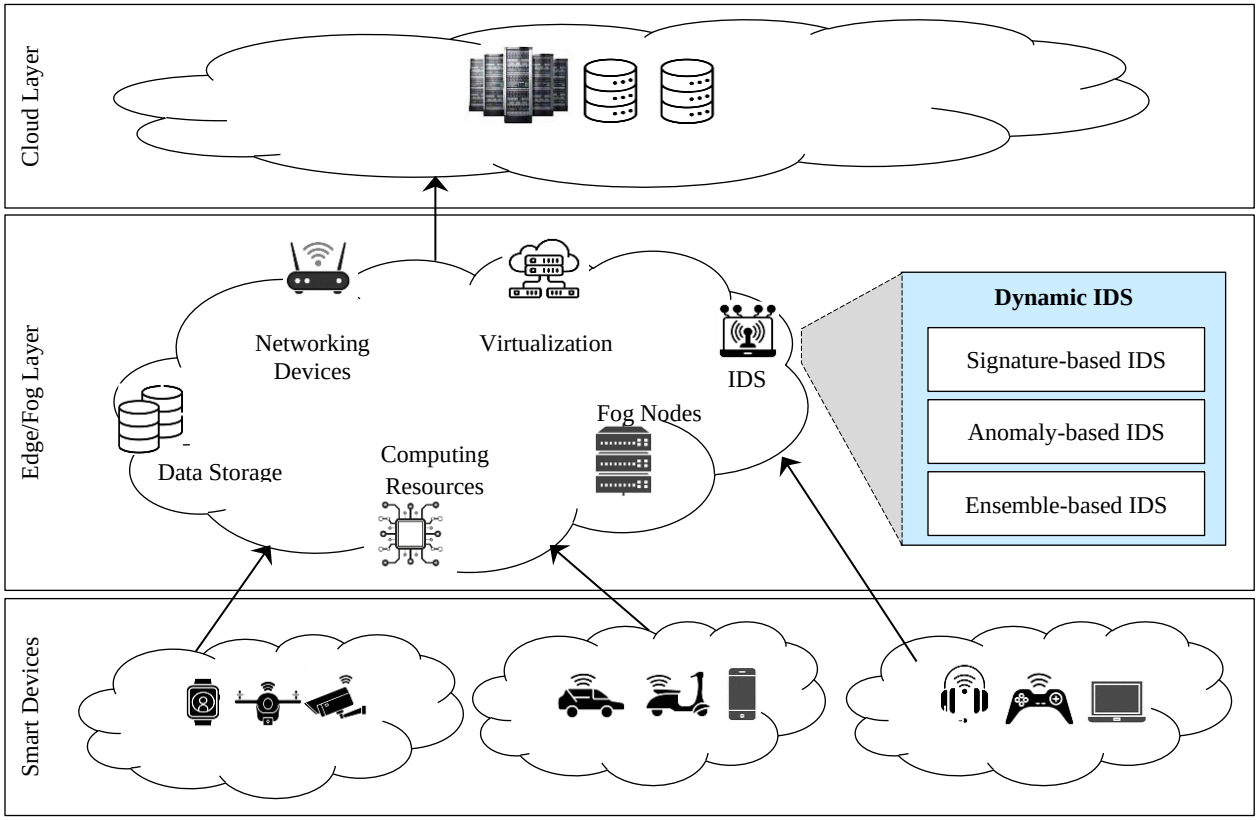

Figure 1 Overview of proposed IDS architecture

The proposed adaptive IDS deployed at the middle layer uses a defence strategy with multiple layers. It includes signature-based and anomaly-based detection methods, as well as ensemble learning techniques that perform behaviour analysis to find intrusion attacks. For FC and EC environments, this adaptive IDS is especially suitable because of its scalability, agility, and security analysis. The detailed framework is presented in *Figure 2*.

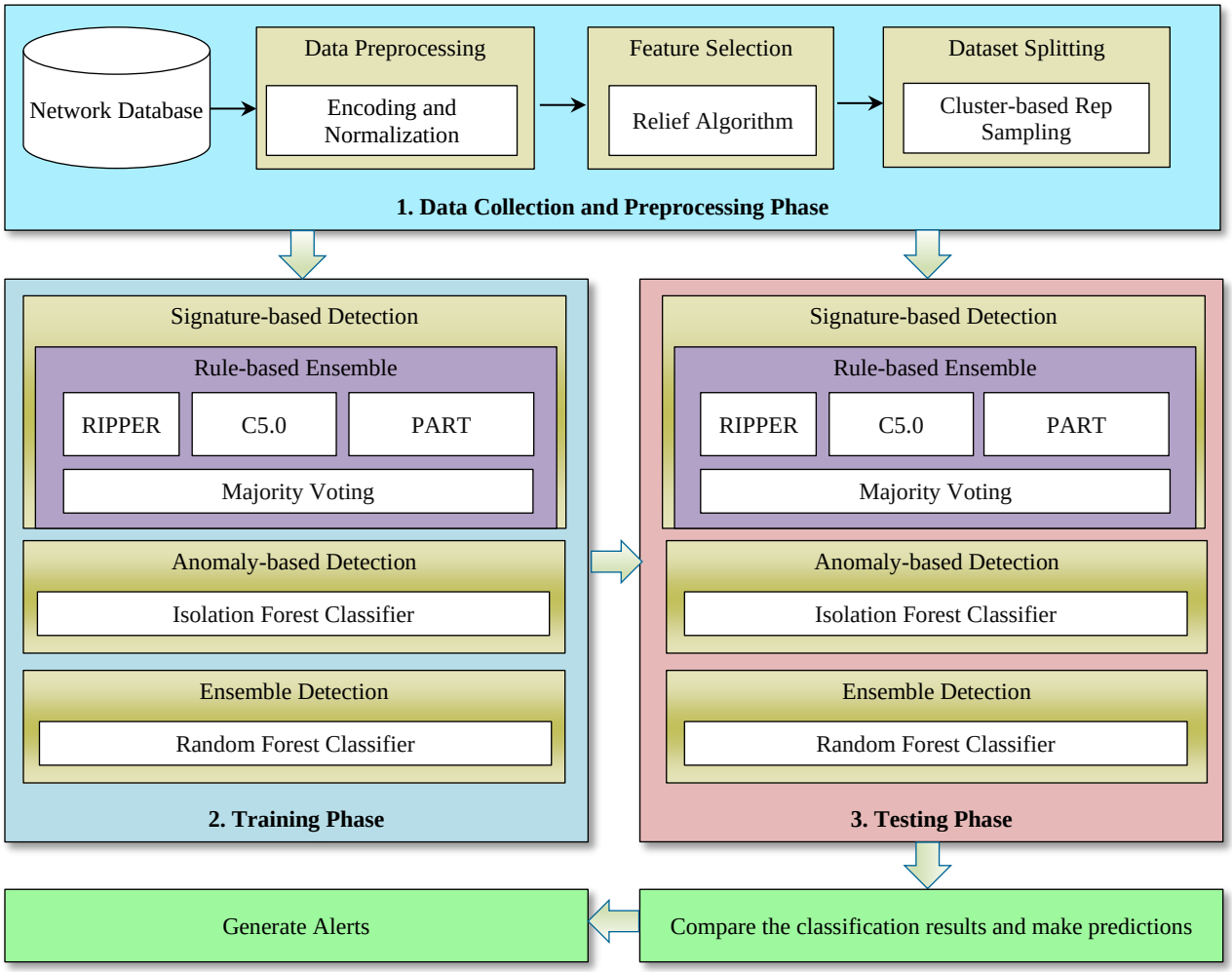

Figure 2 Proposed adaptive IDS in edge/fog environment

For signature-based intrusion detection, the proposed model employs RBE models involving repeated incremental pruning to produce error reduction (RIPPER) classifiers, C5.0 classifiers, and partial decision tree (PART) classifiers, for which the final decision is made using majority voting. For anomaly-based intrusion detection, the IF is used. Further, the third layer includes RF, an ensemble learning technique to decide on the outcome produced by the signature-based and anomaly-based methods. The methodology involved preprocessing and analyzing a significant volume of data from the UNSW-NB15 and NSL-KDD datasets. The UNSW-NB15 dataset, comprising 82,332 instances, was split into 57,632 training and 24,700 testing instances. Similarly, the NSL-KDD dataset, with a total of 25,192 instances, was divided into 17,634 training and 7,558 testing instances. The comprehensive evaluation of the proposed model was bolstered by the robust foundation provided by these large dataset sizes. The details of each phase in the proposed IDS are explained in the subsections below.

3.1Data collection and preprocessing

In this study, the data was collected from publicly available datasets, namely NSL-KDD and UNSW-NB15, which were selected for their relevance in intrusion detection tasks. A detailed description of these datasets is provided in Section 4.2. While the datasets were obtained from established sources, this section outlines a general procedure for data collection to understand the details of the data collection process. Further, the preprocessing steps proposed in the study, which are applicable to both datasets, ensure consistency and reliability in the analysis.

For intrusion detection, the data is generally collected from the network. This collected data includes a variety of features from source to destination, such as internet protocol (IP) address, port used, protocol used, number of bytes transmitted, number of bits per second, packet count, inter-packet arrival time, and more. After the data is gathered, preprocessing steps are applied to remove duplicate or missing records. Next, categorical attribute values are encoded into numerical values. Various encoding techniques like binary encoding, ordinal encoding, and one-hot encoding can be used, but the proposed model employs one-hot encoding, as it is widely used and has been shown to provide effective performance in classification tasks [25, 53]. Additionally, computational complexity may increase due to the large-scale attributes in certain features. To address this, features with higher attribute values are scaled from 0 to 1 using min-max normalization [54]. After preprocessing, the next step is to annotate the dataset according to sample behavior. This involves tagging the data with relevant information to help the system better understand it.

The two key processes before classification are feature selection and dataset splitting. In the proposed model, the Relief feature selection algorithm is used for feature selection. This ML technique identifies the most relevant features in a dataset by evaluating their relevance to the target variable [55]. The Relief model assigns weights to individual features based on their ability to distinguish between similar and dissimilar instances. The algorithm begins by setting all feature weights to zero, then randomly selects instances from the dataset and calculates distances between them for each feature. The feature weights are updated by comparing the nearest neighbors of the selected instance within the same class (nearest hit) and those of a different class (nearest miss). Features that significantly help distinguish between similar and dissimilar instances are given higher weights. The algorithm iterates until convergence and ranks the features according to their weights, with higher weights indicating greater relevance. The Relief algorithm is effective in handling mixed-feature datasets and noisy data.

3.1.1Cluster-based representative sampling (CRS)

This subsection presents the detailed procedure of CRS for data splitting. In this CRS approach, the instances are initially divided into subsets based on their class labels. This ensures that the instances belonging to the same class are grouped, which helps in capturing the unique properties of each class. Once the instances are grouped based on their class labels, the labels are removed before applying clustering to avoid bias and ensure that clusters are formed solely based on the inherent patterns of the data. The clustering process is employed inside each class group to cluster the data into subsets that are distinguished by similar patterns or characteristics. For each class group, k-means clustering is applied to classify distinct groups of instances within their respective classes. Here, the elbow method is used to identify the optimal k value. It identifies the point in a plot of the within-cluster sum of squares where the rate of decrease slows down, resembling an 'elbow' [56]. The elbow method aids in selecting the optimal k by identifying the point where increasing clusters results in diminishing variance reduction returns.

After creating clusters for each class, the next step is to select representative samples from each cluster. Representatives are selected according to specified criteria, including closeness to the cluster centre or capacity to adequately reflect the cluster's properties. The selected samples are chosen to accurately represent the diversity within each cluster. Around 70% of the representative data from each cluster is selected for the training set, and the remaining is used for the test set. A well-balanced representation of all classes is ensured in the training set, which comprises 70% of the representative samples from each class. The remaining 30% of the data from each class is used for a test set to assess the efficiency of the model. This method effectively captures the diversity within classes, which is crucial for building a model that generalizes well. Additionally, it helps manage computational complexity by focusing on representative samples rather than the entire dataset. Selecting representative samples from each cluster allows for training on data with finer details, which results in a more accurate and resilient model. These steps are summarized in Algorithm 1, which provides a detailed procedural outline of the CRS method.

Algorithm 1. CRS for Data Splitting

Input: Dataset D with n instances with m classes
Output: training set and test set
Procedure CRS_datasplitting()
Begin
    Partition D into m subsets based on their class labels.
    For each class group i in m:
        Apply K-means clustering to the instances
        Use the Elbow method to determine the optimal k
        Perform K-means clustering with k clusters.
        For each cluster k:
            Select 70% representative samples
            Add representative samples to class_train
            Allocate remaining samples to class_test

End For
Add the class_train set to the final training set.
Add the class_test set to the final test set.
End For
Return the final training and test sets
End Procedure

3.2Proposed adaptive IDS classification model

3.2.1Signature-based intrusion detection module

Signature-based intrusion detection is a powerful method for detecting known attack patterns. Generally, rule-based classifiers or pattern matching are widely used to detect known attack patterns within network traffic. The rule-based classifiers are trained with a set of predefined patterns that are specific to various known attacks. The incoming traffic is identified as a specific attack when it meets these established rules or criteria. The proposed method adopts an ensemble of rule-based classifier methods instead of a single rule-based classifier. Ensemble methods improve signature-based detection accuracy by combining multiple rule-based models and utilizing their numerous generated rules, or DTs, based on network traffic features.

The set of models employed in RBE includes RIPPER [21], C5.0 [50], and PART [30] classifiers.

- *RIPPER:* A rule-based classifier that generates simple, comprehensible rules from training data, which helps in detecting well-defined attack patterns. The rules are based on constraints and are useful for straightforward pattern matching.
- *C5.0:* An algorithm that builds DTs to classify instances. C5.0 enhances DTs by refining splitting criteria, enabling more precise rules and effective classification of complex attack patterns.
- *PART:* Constructs partial DTs that focus on generating rules for different subsets of data. This approach supports the construction of specific rules that can complement those generated by other classifiers.

One typical ensemble strategy that uses the voting of the individual classifiers to determine the final label is majority voting [54], which is employed in this method to make the final prediction from these classifiers. This voting method improves detection accuracy by combining predictions from multiple classifiers, reducing FPs and FNs by leveraging the strengths of each classifier, thereby increasing the true positives (TPs) and true negatives (TNs). The generated rules from these classifiers are stored in the database during the training phase, and the incoming network data is compared with these rules. If an attack is identified, an alert is generated. Furthermore, the rules are updated periodically to adapt to evolving attack patterns, ensuring that the IDS remains effective against new and emerging threats. The algorithm pseudocode for signature-based intrusion detection is given in Algorithm 2.

Algorithm 2. Signature-based Intrusion Detection
Input: Network traffic data for training and testing
Output: Alerts for detected attacks
Procedure SignatureBasedDetection()
Begin
Initialize an empty rule database.
Train rule-based ensemble classifiers:
Train RIPPER: Generate simple rules.
Train C5.0: Build DT to classify instances.
Train PART: Construct partial DTs.
Store all the rule groups.
For each test data from network traffic:
Compare the data with stored rule groups.
Combine the predictions using a majority vote
Generate alerts based on ensemble prediction.
End For
Update the rule groups periodically.
End Procedure

3.2.2Anomaly-based intrusion detection module

Anomaly-based intrusion detection complements signature-based systems by detecting unknown or novel attack patterns that do not match known signatures. It offers a flexible approach that can adapt to new threats and changes in network behaviour. The proposed model employs an IF for detecting anomalies in network traffic [24, 28]. Using a random feature selection process, the IF algorithm isolates anomalies by recursively partitioning the data. Each split operation isolates observations, and the depth of the isolation tree helps determine the anomaly score. This method is effective because anomalies are typically isolated more quickly than normal observations. The random selection of features and split values helps ensure that the model is not biased by any particular feature and captures various patterns within the data. The isolation process continues until each data point is isolated, making the algorithm well-suited for detecting anomalies. Anomaly scores are computed based on the depth of isolation. Points that require fewer splits to be isolated are considered anomalies. This approach helps distinguish between normal and anomalous data points effectively. The IF model is trained on network traffic data to learn the typical patterns and identify outliers. Periodic updates are crucial to maintaining the accuracy of the detection system as new patterns and attack methods emerge. A predefined threshold is used to decide when an anomaly alert should be triggered. Setting an

appropriate threshold is essential to balancing FPs and FNs. This anomaly-based detection process is summarized as algorithm pseudocode in Algorithm 3.

Algorithm 3. Anomaly-based Intrusion Detection

```
Input: Network traffic data for training and testing
Output: Alerts for anomalies
Procedure AnamolybasedDetection()
Begin
    Initialize an empty forest of isolation trees.
    For each tree in the forest:
        Randomly select a subset of the train_data
        Construct an isolation tree:
        Do
            Choose a random feature and split the value.
            Partition the data into two subsets.
        While the termination condition is not met
    End For
    Initialize anomaly threshold.
    While true do
        Receive incoming network data.
        For each data point:
            Compute avg depth of a point in all trees
            Calculate the anomaly score
            If the anomaly score exceeds the threshold,
then
                Generate alert for anomaly.
            End If
        End For
    End While
End Procedure
```

3.2.3Ensemble learning technique

Though signature-based and anomaly-based intrusion detection methods identify known and unknown attacks efficiently, these methods may generate FNs for unknown attacks or FPs for minor deviations in normal behaviour, respectively. To address these issues, the proposed model employs an ensemble-based learning technique that combines multiple ML models to improve predictive performance compared to individual classifiers. It is a powerful ML method that employs diverse classification models to make more accurate predictions, thereby improving robustness and reducing overfitting. More specifically, an RF is employed, which is suitable for both classification and regression problems. It works by constructing a multitude of DTs during the training phase and classifying the test data based on a voting mechanism. It uses bootstrapped sampling or bagging, which selects a subset of the training data with replacement. The RF model is chosen for the proposed model since it has been proven to be effective in classifying normal and anomalous network traffic data [49].

The RF model is constructed using a set of training instances that include both normal and malicious network samples. As new data arrives, the model classifies the test sample using the ensemble of DTs. The outputs of the RF detection methods help in making the final decision by comparing the results for the sample with those from the signature-based and anomaly-based intrusion detection models. If the RF identifies the network traffic as an attack and either one or both of the signature-based and anomaly-based intrusion detection models detect it as an attack, then it signifies that it is a serious attack, and thus an alert is generated. If the RF identifies the input sample as an attack and none of the signature-based and anomaly-based intrusion detection models identify it as an attack, then the sample seems suspicious. Furthermore, if both the signature-based and anomaly-based intrusion detection models detect it as an attack, but the RF classifies it as normal, it also signifies suspicion. On the other hand, if either the signature-based or anomaly-based intrusion detection models detect it as an attack and the RF detects it as normal, then it is an FP, and no alert is generated.

Thus, the outputs of the signature-based and anomaly-based models act as feedback for the ensemble model and provide valuable insights. Furthermore, the models are updated dynamically by adjusting decision boundaries and decision rules. With the updated behaviour, the models classify incoming samples as normal or malicious by utilizing collective insights from signature-based, anomaly-based, and ensemble learning-based detections. Upon predicting the input samples as malicious, the system generates an alert, enabling a timely response to potential security threats. Incorporating signature-based, anomaly-based, and ensemble learning techniques into a single detection model maximizes the model's performance through complementary strengths and a diversity of decision mechanisms. The algorithm pseudocode for the ensemble learning technique is presented in Algorithm 4.

Algorithm 4. Ensemble Learning based Intrusion Detection

```
Input: Network traffic data for training and testing
Output: Alerts for anomalies
Procedure EsembleLearningModel()
Begin
    Load training data.
    Initialize an empty RF model.
    Train RF model using training data
    While true do
        Receive new traffic data
```

```
    Apply signature-based detection (SD)
    Apply anomaly-based detection (AD)
    Classify new data using an RF classifier
    For each network sample NS in test data:
      If RF classifies the NS as an attack, then
        If NS=attack for SD or AD then
            Generate an alert for a serious attack.
        Else If NS==normal for SD or AD then
            Generate a suspicious alert
        End If
      Else if RF classifies the NS as normal then
        If NS==attack for SD and AD then
            Generate a suspicious alert
        Else If NS==attack for SD or AD then
            Mark it as a false positive
        End If
      End If
    End For
  End While
End Procedure
```

4.Results

4.1Experimental setup

This section presents the experimental results and analysis for the proposed adaptive IDS for FC and EC environments to protect against various cyberattacks. The source code for various ML models is written in the Python programming language, utilizing its built-in libraries and TensorFlow. It is installed on an Intel(R) Pentium(R) machine with a CPU 6405U at 2.40 GHz and 8 GB of RAM running a 64-bit Windows operating system. The experiments for the proposed and other models are evaluated using two publicly available network datasets: UNSW-NB15, which contains 82,332 instances, and NSL-KDD, which has 25,192 instances. These datasets were downloaded from well-known platforms, GitHub and Kaggle, which offer a wide range of free public datasets for ML tasks and data science competitions. Furthermore, the implemented IDS framework incorporates basic error-handling protocols to manage system failures. It generates error logs to document issues and triggers automated notifications to administrators for prompt resolution. The system is designed to manage common failures such as data corruption or network disruptions to maintain operational stability. Also, various classifiers such as RIPPER, C5.0, PART, IF, and RF are employed in the study, for which the parameters are tuned using grid search to improve model performance. The parameters used for these classifiers are listed in *Table 1*.

Table 1 Parameters used for various classifiers in the proposed study

Classifier	Parameter Configuration
RIPPER	minNumObj (Minimum number of instances per rule): 5
	pruning (Whether to prune rules): true
C5.0	Trials (Number of boosting trials): 10
	minCases (Minimum number of cases in a leaf): 2
PART	minNumObj (Minimum number of instances in a leaf): 4
IF	n_estimators (Number of base estimators): 150
	contamination (Proportion of outliers in the data): 'auto'
RF	n_estimators (Number of trees in the forest): 200
	max_depth (Maximum depth of each tree): 10
	min_samples_split (Minimum number of samples required to split an internal node): 2

4.2Datasets utilized

Though multiple network traffic datasets are available in the literature, UNSW-NB15 is a modernized network dataset [57]. The IXIA PerfectStorm tool at UNSW Canberra's cyber range lab generated the raw network packets for this dataset to produce a combination of actual present-day activities and synthetic modern attack behaviours. The tcpdump tool was used to record 100 GB of raw traffic, construct twelve algorithms using Argus and Bro-IDS tools, and produce 49 features, which are grouped into five categories: flow, basic, content, time, and additional features related to connection [14]. This dataset is available as two files in the form of comma-separated values (CSV), comprising a training set and a testing set. This dataset combines real and synthetic traffic to accurately reflect current network conditions and attack patterns, which is crucial for developing robust IDS models. The inclusion of modern attack types makes it particularly relevant for contemporary cybersecurity research. Additionally, the dataset's availability in CSV files with distinct training and testing sets facilitates simple integration into ML workflows. These attributes make the UNSW-NB15 dataset an ideal choice for this study.

For effective analysis, the testing set containing various attack labels was used. The dataset was downloaded from the GitHub repository, a popular

online platform for hosting and sharing code, datasets, and other development resources. It involves 82,332 instances across 10 classes, including normal, reconnaissance, backdoor, DoS, exploits, analysis, fuzzers, worms, shellcode, and generic. After preprocessing, the instances are divided into training and testing instances in the ratio of 70%:30%, respectively, using CRS. The description of the UNSW-NB15 dataset, involving 37,000 normal samples and 45,332 attack samples used in the study, is broken down in *Table 2*. The dataset contains a total of 82,332 instances, with 57,632 for training and 24,700 for testing. The table lists the number of training and testing instances for each class, including normal, reconnaissance, backdoor, dos, exploits, analysis, fuzzers, worms, shellcode, and generic, along with the total number of instances for each category. Furthermore, the feature selection process is carried out using the Relief algorithm, and 15 significant features are selected for analysis. The list of features grouped based on the obtained feature scores is presented in *Table 3*, with the bold values representing the features selected for the analysis.

Table 2 UNSW-NB15 dataset description

Class Category	**#Training Instances**	**#Test Instances**	**#Total Instances**
Normal	25908	11092	37000
Reconnaissance	2467	1029	3496
Backdoor	395	188	583
DoS	2862	1227	4089
Exploits	7755	3377	11132
Analysis	469	208	677
Fuzzers	4268	1794	6062
Worms	34	10	44
Shellcode	279	99	378
Generic	13195	5676	18871
Total	*57632*	*24700*	*82332*

Table 3 Feature scores for the UNSW-NB15 dataset

#	**Features**	**Score**
1	**service**	**0.201**
2	**dttl**	**0.148**
3	**sttl**	**0.142**
4	**proto**	**0.100**
5	**ct_dst_sport_ltm**	**0.064**
6	**ct_state_ttl**	**0.058**
7	**state**	**0.053**
8	**ct_srv_dst**	**0.051**
9	**ct_srv_src**	**0.048**
10	**ct_src_ltm**	**0.046**
11	**ct_dst_src_ltm**	**0.045**
12	**ct_dst_ltm**	**0.043**
13	**smean**	**0.043**
14	**ct_src_dport_ltm**	**0.038**
15	**dmean**	**0.037**
16	dload	0.025
17	dtcpb	0.023
18	stcpb	0.022
19	swin	0.020
20	sinpkt	0.012
21	is_sm_ips_ports	0.011
22	sload	0.007
23	tcprtt	0.006
24	Synack, ct_flw_http_mthd, ackdat	0.004

#	Features	Score
25	sjit, ct_ftp_cmd, is_ftp_login	0.002
26	dinpkt, spkts, djit, dpkts, dloss, dstip	0.001
28	script, sport, dport, dbytes, sloss, sbytes, trans_depth, dwin, response_body_len, ltime, stime	0.000

The second dataset used for the analysis is the NSL-KDD dataset, which was created to overcome the inherent deficiencies of the popular KDD'99 [58]. Despite some limitations due to a lack of public data for network-based IDSs, this dataset can serve as a valuable benchmark for researchers comparing intrusion detection methods. Since the NSL-KDD training and test sets have a suitable number of records, they provide reliable experiments without random selection and consistent evaluation outcomes. More specific merits of NSL-KDD over KDD'99 are that it eliminates redundant records, ensures no bias towards more frequent records, and guarantees that learners' performance is not affected by methods with better detection rates. Additionally, the number of records in the KDD dataset is proportional to the difficulty level, allowing for a wider range of classification rates for different ML methods. For the analysis, KDDTrain+_20Percent, a CSV file representing a 20% subset of KDDTrain+ with 25,192 instances from a total of 125,973 training instances, was used. The dataset was downloaded from the Kaggle data repository. This NSL-KDD dataset includes 13,449 normal and 11,743 attack instances, totalling 25,192 instances. It is further divided into 17,634 training and 7,558 testing instances. The dataset is categorized into five classes, with four attack classes: DoS, remote-to-local (R2L), probes, and user-to-root (U2R) [59]. A detailed instance distribution is shown in *Table 4*. The four attack classes—DoS, Probes, R2L, and U2R—in the dataset encompass over 20 attack types, which are detailed in *Table 5*.

Table 4 NSL-KDD dataset description

Class category	#Training instances	#Testing instances	#Total instances
Normal	9390	4059	13449
DoS	6463	2771	9234
R2L	146	63	209
Probes	1628	661	2289
U2R	7	4	11
Total	*17634*	*7558*	*25192*

Table 5 Attack Classification in the NSL-KDD dataset

Attack class	Attack types
DoS	Back, Land, Neptune, Pod, Smurf, Teardrop
Probes	Satan, Ipsweep, Nmap, Portsweep
R2L	Guess_Password, Ftp_write, Imap, Phf, Multihop, Warezmaster, Warezclient, Spy
U2R	Buffer_overflow, Loadmodule, Rootkit

Following preprocessing, instances are divided into training and testing instances in a 70%:30% ratio using the CRS algorithm. The dataset reduces duplicates and redundant records, avoiding bias towards frequent attack types, and is proportional to instance difficulty level, enabling a nuanced evaluation of ML methods. Additionally, the dataset includes over 20 attack types categorized into five classes, providing a broad spectrum of scenarios for IDS testing. After preprocessing, the Relief algorithm is used for feature selection, resulting in 17 significant features for analysis, which are grouped based on scores and displayed in *Table 6*, with bold values representing selected features for the analysis.

The UNSW-NB15 and NSL-KDD datasets were chosen for their precise representation of cyber threats and network traffic, as well as their support for in-depth statistical studies of intrusion detection methods. These datasets help validate the proposed IDS framework's effectiveness, analyze significant attributes for intrusion detection, and provide insights into the framework's generalizability across different network environments.

Table 6 Feature scores for NSL-KDD dataset

#	Features	Score
1	**service**	**0.630**
2	**same_srv_rate**	**0.229**
3	**flag**	**0.214**
4	**dst_host_count**	**0.189**
5	**dst_host_same_src_port_rate**	**0.189**
6	**dst_host_serror_rate**	**0.168**
7	**logged_in**	**0.164**
8	**dst_host_srv_count**	**0.146**
9	**protocol_type**	**0.128**
10	**dst_host_same_srv_rate**	**0.120**
11	**dst_host_srv_serror_rate**	**0.118**
12	**count**	**0.115**
13	**srv_serror_rate**	**0.082**
14	**dst_host_diff_srv_rate**	**0.075**
15	**dst_host_rerror_rate**	**0.075**
16	**diff_srv_rate**	**0.063**
17	**srv_diff_host_rate**	**0.059**
18	wrong_fragment	0.040
19	srv_count	0.032
20	serror_rate	0.026
21	dst_host_srv_diff_host_rate	0.021
22	rerror_rate	0.017
23	dst_host_srv_rerror_rate	0.015
24	srv_rerror_rate	0.012
25	hot, is_guest_login	0.010
28	duration	0.006
29	dst_bytes, root_shell, num_failed_logins	0.002
30	num_compromised, su_attempted, num_access_files, num_shells, num_root, src_bytes, num_file_creations, urgent, land, num_outbound_cmd, is_host_login	0.000

4.3Evaluation metrics used

The proposed adaptive IDS is evaluated based on common evaluation metrics calculated from the confusion matrix. The confusion matrix is a table with four basic elements: TP, TN, FP, and FN. However, multiclass classification evaluates the performance of a classification model on test data with known labels, helping identify model errors [60].

- TP: The number of correctly predicted positive instances that belong to the positive class.
- TN: The number of correctly predicted negative instances that do not belong to the positive class.
- FP: The number of instances that were incorrectly predicted as positive and belong to the negative class.
- FN: The number of instances that were incorrectly predicted as negative but belong to the positive class.

The following are the evaluation metrics used to evaluate the performance of various models in the proposed study [41]:

Accuracy: It is the percentage of correct classifications. This refers to the ratio of accurate predictions out of the total number of predictions (N) made on the test data containing C classes, as given in Equation 1.

$$Accuracy = \frac{1}{N}\sum_{i=1}^{C} TP_i \quad (1)$$

Precision: It measures the accuracy of positive predictions. This refers to the ratio of TPs of class i out of all instances predicted as positives by the classifier for that class, as given in Equation 2.

$$Precision_i = \frac{TP_i}{TP_i+FP_i} \quad (2)$$

Recall: It measures the model's ability to classify all the positive classes. This refers to the ratio of TPs in class i out of all positive instances for that class, as given in Equation 3.

$$Recall_i = \frac{TP_i}{TP_i+FN_i} \quad (3)$$

F-Measure: It is the harmonic mean of precision and recall that evaluates the balance between precision and recall, as given in Equation 4.

$$F-Measure_i = \frac{2\times recall_i \times precision_i}{recall_i+precision_i} \quad (4)$$

Average accuracy: It is the average recall of all the classes of the test data, as shown in Equation 5.

$$Avgerage\ Accuracy = \frac{1}{C}\sum_{i=1}^{C} Recall_i \quad (5)$$

Attack accuracy: It refers to the model's ability to identify different types of attacks by computing average accuracy without considering normal class, as shown in Equation 6.

$$Attack\ Accuracy = \frac{1}{C-1}\sum_{i=2}^{C} Recall_i \quad (6)$$

Mean F-Measure (MFM): It is calculated as the average F-measure of all the classes of the test data as given in Equation 7.

$$MFM = \frac{1}{C}\sum_{i=1}^{C} F - measure_i \quad (7)$$

ADR: It refers to the rate of accuracy of the attack classes alone, as shown in Equation 8.

$$ADR = \frac{\sum_{i=2}^{C} TP_i}{\sum_{i=2}^{C}(TP_i+FP_i)} \quad (8)$$

FAR: It refers to the rate of normal traffic that is incorrectly predicted as attacks out of all normal traffic. This is computed for the normal class (i = 1) alone using FN (wrong predictions as attacks) and TP (correct predictions as normal traffic), as given in Equation 9.

$$FAR = \frac{FN_1}{TP_1+FN_1} \quad (9)$$

4.4Result analysis

This section presents the results of the proposed and various standard classifiers for the aforementioned network traffic datasets. The results for the proposed RBE for signature-based intrusion detection, IF for anomaly-based intrusion detection, and the combination of models with RF are presented as a C×C confusion matrix. The model's performance was tested on the UNSW-NB15 dataset, comprising 24,700 testing instances, and the NSL-KDD dataset, with 7,558 testing instances. The large volume of data used in the testing phase signifies the robustness and reliability of the results. A detailed analysis for each layer of defence in the proposed adaptive IDS model is evaluated using UNSW-NB15 test data of size 24,700 instances, presented in *Table 7*. The results of the RBE in the form of a confusion matrix for the 10 classes of UNSW-NB15 test data and the class-wise precision, recall, and F-measure are shown in *Table 7*, with grey cells indicating the TPs. The RBE has higher precision for normal (97.12%), reconnaissance (93.01%), analysis (97.47%), and generic (98.51%) classes, but provides lower performance with a backdoor (recall: 50%), DoS (precision: 49.27%), analysis (recall: 37.02%), worms (recall: 50%), and Shellcode (precision: 55.36%) classes. The evaluation of RBE on various classes of UNSW-NB15 test data using precision and recall is also visually depicted in *Figure 3*.

Table 7 Confusion matrix of RBE on UNSW-NB15 test data

Class labels	Normal	reconnaissance	Backdoor	DoS	Exploits	Analysis	Fuzzers	Worms	Shellcode	Generic
Normal	10855	9	0	21	90	0	101	0	10	6
Reconnaissance	10	878	1	70	56	0	8	0	4	2
Backdoor	2	2	94	3	47	0	38	0	0	2
DoS	35	6	2	875	197	0	83	0	12	17
Exploits	50	41	6	447	2613	1	150	2	16	51
Analysis	0	0	0	33	59	77	39	0	0	0
Fuzzers	199	5	6	98	150	0	1327	0	4	5
Worms	0	0	0	0	2	0	2	5	0	1
Shellcode	19	3	0	1	8	0	6	0	62	0
Generic	7	0	2	22	60	1	11	0	4	5569
Precision (%)	*97.12*	*93.01*	*84.68*	*49.27*	*75.04*	*97.47*	*75.18*	*71.43*	*55.36*	*98.51*
Recall (%)	*97.86*	*85.33*	*50.00*	*55.01*	*77.38*	*37.02*	*73.97*	*50.00*	*62.63*	*98.11*
F-measure (%)	*97.49*	*89.00*	*62.88*	*51.98*	*76.19*	*53.66*	*74.57*	*58.82*	*58.77*	*98.31*

Table 8 shows a summary of the performance metrics for the proposed RBE as well as the other models (RIPPER, C5.0, and PART) that were used in the ensemble. The results indicate that RIPPER has a lower FAR of 1.71%, a lower attack accuracy of 38.80%, and an ADR of 65.31%. Though PART has a higher accuracy of 86.49% and an ADR of 78.43%, it has a lower attack accuracy of 50.81%. On the other hand, C5.0 has better average accuracy, attack accuracy, and ADR of 60.73%, 56.75%, and 77.25%, respectively. However, the proposed RBE has superior performance metrics, with an accuracy of 89.70%, an average accuracy of 98.73%, an attack accuracy of 65.49%, an ADR of 83.04%, and a FAR of 2.14%.

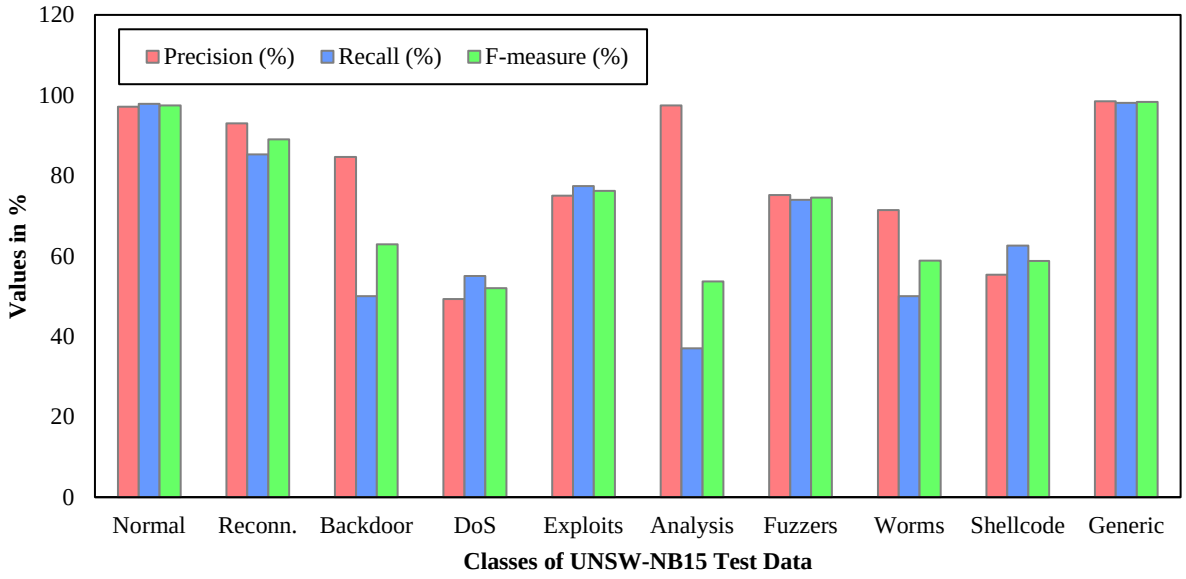

Figure 3 Evaluation of RBE on UNSW-NB15 test data

Table 8 Performance metrics summary for RBE

Metrics	RIPPER	C5.0	PART	RBE
Accuracy	80.12	85.92	86.89	**89.7**
Average accuracy	44.75	60.73	55.46	**68.73**
Attack accuracy	38.8	56.75	50.81	**65.49**
MFM	48.6	59.04	56.95	**72.17**
ADR	65.31	77.25	78.43	**83.04**
FAR	1.71	3.49	2.73	**2.14**

The evaluation of the second layer of defence, the anomaly-based intrusion detection model using IF, is carried out. In this layer of defence, all the labels of the nine attack classes are replaced with a single "attack" label. Combining all attack classes into a single "attack" class simplifies the analysis by focusing on distinguishing between normal and anomalous behaviour, rather than handling multiple specific attack categories. This approach streamlines anomaly detection and emphasizes the system's ability to identify any deviation from normal behaviour. The results, in the form of a confusion matrix for the two classes (normal, attack) of the UNSW-NB15 test data, which consists of 24,700 instances, and the class-wise precision, recall, and F-measure, are shown in *Table 9*, with grey cells indicating the TPs. The results show improved precision (97.20%) for the attack class, better recall (97.17%) for the normal class, and better F-measure for both the normal (87.76%) and attack (87.90%) classes.

Table 9 Confusion matrix of IF on UNSW-NB15 test data

Class labels	Normal	attack	Precision (%)	recall (%)	F-measure (%)
Normal	10778	314	*80.01*	*97.17*	*87.76*
Attack	2692	10916	*97.20*	*80.22*	*87.90*

The overall summary of performance metrics of the IF in detecting anomaly-based intrusion is compared with similar models such as NB and OC-SVM, and the results are presented in *Table 10*. The results indicate that the NB classifier has better performance; however, it has a higher FAR of 6.93%. Although OC-SVM has better performance than the NB classifier, its performance is low when compared with the IF, which has an average attack accuracy of 88.69% and an ADR of 80.22%. Its average accuracy, attack accuracy, and MFM are far greater than those of the RBE model, even though its FAR is 2.83% higher. The evaluation of the proposed adaptive IDS model, including the third layer of defence using RF, is carried out. The results, in the form of a confusion matrix for the 10 classes of

UNSW-NB15 test data, along with the class-wise precision, recall, and F-measure, are shown in *Table 11*, with grey cells indicating the TPs. The confusion matrix is a crucial tool for assessing the effectiveness of an IDS in detecting and classifying attacks, identifying strengths and weaknesses, and benchmarking against existing solutions. The results indicate that the proposed adaptive IDS exhibits strong performance, particularly for normal (precision: 97.66%, recall: 98.47%, F-measure: 98.06%), reconnaissance (precision: 93.17%, recall: 87.56%, F-measure: 90.28%), and generic traffic (precision: 98.52%, recall: 98.63%, F-measure: 98.57%). However, it shows lower efficacy in identifying analysis (recall: 41.83%; F-measure: 58.89%), worms (recall: 50%; F-measure: 58.82%), and shellcode attacks (precision: 58.68%, F-measure: 64.55%). Nonetheless, it performs better than the RBE and anomaly-based detection methods. The evaluation of the proposed adaptive IDS with three defence layers on various classes of UNSW-NB15 test data using different performance metrics is depicted in *Figure 4*. The results indicate that the detection of all classes of attacks has notably increased, indicating improved performance.

Table 10 Summary of performance metrics for IF

Metrics	NB	OC-SVM	IF
Accuracy	84.04	86.84	**87.83**
Average accuracy	84.88	87.63	**88.69**
Attack accuracy	76.69	79.89	**80.22**
MFM	84.04	86.84	**87.83**
ADR	76.69	79.89	**80.22**
FAR	6.93	4.63	**2.83**

Table 11 Confusion matrix of proposed adaptive IDS on UNSW-NB15 test data

Class Labels	Normal	Reconnaissance	Backdoor	DoS	Exploits	Analysis	Fuzzers	Worms	Shellcode	Generic
Normal	10922	9	0	21	66	0	58	0	10	6
Reconnaissance	10	901	1	51	52	0	8	0	4	2
Backdoor	2	2	117	3	31	0	31	0	0	2
DoS	35	6	2	945	127	0	83	0	12	17
Exploits	50	41	6	217	2843	1	150	2	16	51
Analysis	0	0	0	33	47	87	41	0	0	0
Fuzzers	148	5	5	96	128	0	1403	0	4	5
Worms	0	0	0	0	2	0	2	5	0	1
Shellcode	10	3	0	1	8	0	6	0	71	0
Generic	7	0	2	22	32	1	10	0	4	5598
Precision (%)	*97.66*	*93.17*	*87.97*	*68.03*	*85.22*	*97.75*	*78.73*	*71.43*	*58.68*	*98.52*
Recall (%)	*98.47*	*87.56*	*62.23*	*77.68*	*84.19*	*41.83*	*78.21*	*50.00*	*71.72*	*98.63*
F-measure (%)	*98.06*	*90.28*	*72.90*	*72.52*	*84.70*	*58.59*	*78.47*	*58.82*	*64.55*	*98.57*

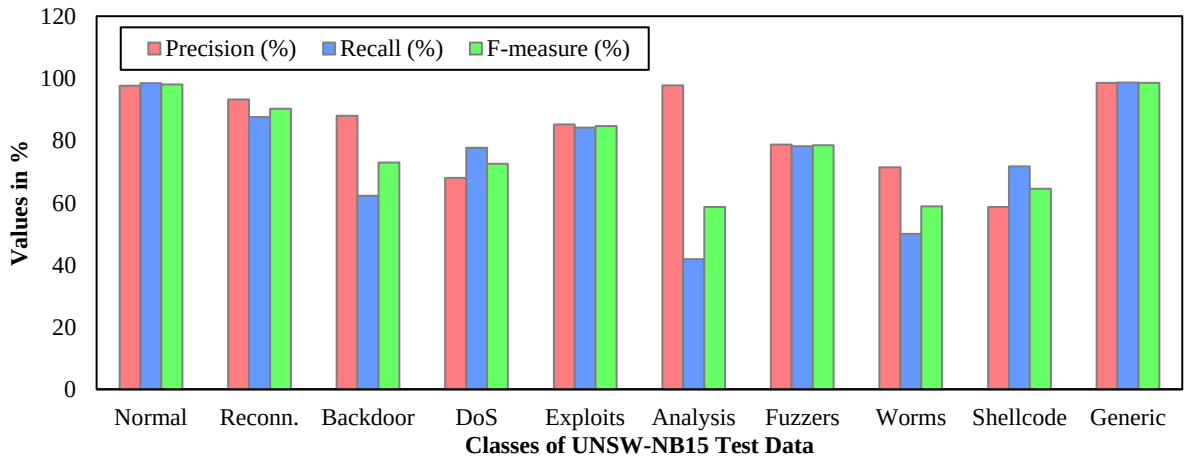

Figure 4 Evaluation of the proposed model on UNSW-NB15 test data

The overall summary of performance metrics for the proposed adaptive IDS with RF and the other individual models, such as MNC, AB, XGB, SVM, MLP, and CART, is presented in *Table 12*. The results indicate that the MNC and SVM classifiers have lower performance than the other classifiers under comparison. Although AB and CART have better accuracy rates of 87.12% and 86.86%, respectively, they have lower average accuracy and attack accuracy. The XGB has a better accuracy of 87.19% but a lower average accuracy of 60.57% and an attack accuracy of 56.44%. Among all the classifiers, MLP has the highest accuracy of 89.02%, an MFM of 70.88%, an ADR of 82.24%, and a lower FAR of 2.22%. However, the proposed adaptive IDS with RBE, IF, and RF algorithms has a higher accuracy of 92.68%, an average accuracy of 75.05%, an attack accuracy of 72.45%, and an MFM of 77.75%, with a lower FAR of 1.53%.

Table 12 Summary of performance metrics for proposed adaptive IDS

Classifiers	Accuracy	Average accuracy	Attack accuracy	MFM	ADR	FAR
MNC	81.93	54.17	49.57	61.49	70.81	4.44
AB	87.12	54.87	50.08	56.98	78.28	2.04
XGB	87.19	60.57	56.44	63.94	78.90	2.29
SVM	81.33	50.38	45.16	57.87	68.22	2.70
MLP	89.02	67.95	64.64	70.88	82.24	2.22
CART	86.86	54.71	49.92	57.05	77.89	2.14
Adaptive IDS	**92.68**	**75.05**	**72.45**	**77.75**	**88.03**	**1.53**

For easy interpretation of the results obtained for the UNSW-NB15 test data from signature-based detection using RBE, anomaly-based detection using IF, and adaptive ensemble-based IDS in a three-layer defence, the normal and attack classes are assessed using precision, recall, and F-measure, as depicted in *Figure 5*. The adaptive IDS demonstrates impressive performance in predicting normal traffic (precision: 97.66%; recall: 98.47%; F-measure: 98.06%), effectively minimizing FPs. Although anomaly-based detection performs better in detecting attack traffic (precision: 97.20%; recall: 80.02%; F-measure: 87.90%), its performance with normal traffic is notably low (precision: 80.01%; recall: 97.17%; F-measure: 87.86%). Moreover, when comparing the proposed model with and without the feature selection method (Relief and CRS), it is clear that the proposed model with feature selection has higher performance for both normal (precision: 97.66%; recall: 98.47%; F-measure: 98.06%) and attack (precision: 82.17%; recall: 72.45%; F-measure: 75.49%) classes than the proposed model without feature selection using Relief and CRS for normal (precision: 88.18%; recall: 89.01%; F-measure: 88.59%) and attack (precision: 79.11%; recall: 68.32%; F-measure: 73.32%) classes. Thus, the adaptive IDS stands out due to its high precision and recall for normal traffic, making it a reliable choice for environments where minimizing FPs is critical.

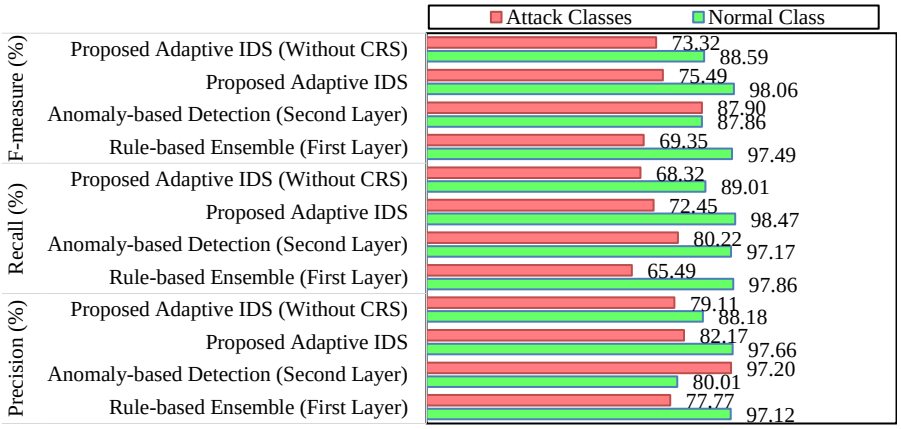

Figure 5 Comparative analysis for different layers of IDS

The evaluation has been conducted for the proposed adaptive IDS model, combining the results of RBE for signature-based intrusion detection, IF for anomaly-based intrusion detection, and RF using the NSL-KDD dataset, which has 25,192 instances. The results of the proposed model are presented as a C×C confusion matrix for 5 classes of NSL-KDD test data with 7,558 instances, along with the class-wise precision, recall, and F-measure shown in *Table 13*, with grey cells indicating the TPs. The results indicate that the method has the best performance with the normal, DoS, and Probes classes, which are generally well-represented in the training data. Despite having high precision (95%), the method has a lower recall (90.48%) for the R2L class, indicating that it may miss some R2L attacks or misclassify them as other types. Additionally, the method's high precision (100%) but low recall (50%) suggests it only detects a portion of U2R attacks, resulting in high FNs. The evaluation of the proposed adaptive IDS with three defence layers on various classes of NSL-KDD test data using precision, recall, and F-measure is depicted in *Figure 6*.

Table 13 Confusion matrix of proposed adaptive IDS on NSL-KDD test data

Class Labels	Normal	DoS	R2L	Probes	U2R
Normal	4049	7	5	3	1
DoS	4	2761	0	5	1
R2L	2	0	57	1	0
Probes	4	3	1	652	0
U2R	0	0	0	0	2
Precision (%)	99.61	99.64	95	98.79	100
Recall (%)	99.75	99.64	90.48	98.64	50
F-Measure	99.68	99.64	92.68	98.71	66.67

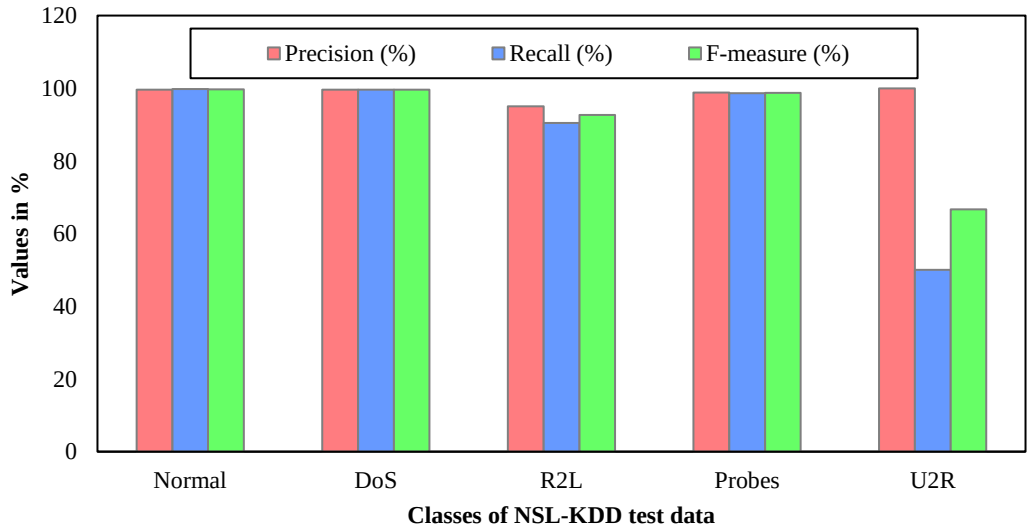

Figure 6 Evaluation of the proposed model on NSL-KDD test data

The overall summary of performance metrics for the proposed adaptive IDS and the other individual models used in the proposed model, such as RIPPER, C5.0, PART, RBE, and IF, is presented in *Table 14*. The results indicate that the C5.0 classifier shows a higher average accuracy of 80.15% and an attack accuracy of 75.45% compared to the RIPPER and PART classifiers. PART has a higher accuracy of 98.98% with a lower FAR of 0.74%, but lower average and attack accuracies of 79.47% and 74.52%, respectively. The IF has a higher attack accuracy of 82.22% and a higher MFM of 89.24% than the other models under comparison. However, the proposed adaptive IDS has a higher accuracy of 99.51%, an average accuracy of 87.70%, an attack accuracy of 84.69%, and an ADR of 99.23%, with a lower FAR of 0.25%.

Table 14 Performance summary for proposed adaptive IDS on NSL-KDD test data

Classifiers	Accuracy	Average accuracy	Attack accuracy	MFM	ADR	FAR
RIPPER	98.84	79.33	74.37	80.92	98.46	0.84
C5.0	98.72	80.15	75.45	80.62	98.49	1.08
PART	98.98	79.47	74.52	83.59	98.66	0.74
RBE	98.89	85.26	81.78	88.89	98.57	0.84
IF	98.95	85.62	82.22	89.24	98.66	0.79
Proposed	**99.51**	**87.70**	**84.69**	**91.48**	**99.23**	**0.25**

4.5Comparative analysis

The performance of the proposed adaptive IDS is compared with various state-of-the-art intrusion detection models from the literature. The comparison based on the UNSW-NB15 dataset is listed in *Table 15*. From the analysis, it is clear that the proposed adaptive IDS has a better accuracy of 92.68% than various other models from different studies. The unified IDS [51] exhibit superior performance, with average accuracy, attack accuracy, and MFM of 77.87%, 73.28%, and 79.12%, respectively. Furthermore, EHIDS [14] has the lowest FAR of 1.10%. However, the superiority of the proposed model lies in its highest ADR of 88.03%. The comparison of the proposed model with existing models based on the NSL-KDD dataset is also listed in *Table 15*. From the analysis, it is evident that the proposed adaptive IDS has a better accuracy of 92.68% and an MFM of 91.48% than various other models from different studies. Moreover, the DENDRON model [41] shows improved performance with an average accuracy of 90.54% and an attack accuracy of 88.44%. The C5.0 has an average accuracy of 88.09% and attack accuracy of 85.23%. However, the superiority of the proposed model lies in its highest ADR of 99.23% and lowest FAR of 0.25%.

Table 15 Performance comparison with other models

Authors	Other models	Accuracy	MFM	Average accuracy	Attack accuracy	ADR	FAR
UNSW-NB15 dataset							
Moustafa and Slay (2016) [42]	DT	85.56	-	-	-	-	15.78
Papamartzivanos et al. (2018) [41]	DENDRON	84.33	48.81	52.21	47.19	63.76	2.61
	C4.5	85.15	48.79	49.33	44.14	67.88	2.54
Almogren (2020) [38]	DBN	73.70	34.47	38.76	35.03	74.88	27.64
Kumar et al. (2021) [23]	UIDS	88.92	**79.12**	**77.87**	**73.29**	86.15	7.18
	NN	86.70	68.04	71.20	64.80	66.40	3.40
	C5.0	89.76	75.10	74.50	68.75	81.76	1.68
	SVM	78.77	62.18	61.18	53.27	72.70	7.18
Kumar et al. (2020) [43]	C5.0	90.74	75.54	75.80	70.65	83.47	3.70
	Integrated Rule	84.83	68.13	65.21	57.01	90.32	2.01
Singh et al. (2022) [14]	EHIDF	90.25	74.35	70.54	67.46	86.95	**1.10**
Proposed Model	Adaptive IDS	**92.68**	77.75	75.05	72.45	**88.03**	1.53
NSL-KDD dataset							
Panda et al. (2010) [35]	MNB	81.47	-	-	-	-	12.85
Papamartzivanos et al. (2018) [41]	DENDRON	97.55	83.35	**90.54**	**88.44**	95.97	1.08
	C4.5	99.36	86.57	88.09	85.23	99.19	0.26
	NB	68.3	44.77	61.56	62.54	80.55	29.79
	SMO (SVM)	96.14	66.18	64.60	56.07	93.12	0.68
	MLP	97.41	65.96	66.46	58.29	95.42	0.47
	ID3	96.32	69.34	66.36	58.2	93.69	0.54
Khraisat et al. (2020) [18]	Stacking Ensemble	83.24	83.25	75.29	72.70	72.75	2.17
Bakro et al. (2023) [25]	Ensemble	990.1	-	-	-	-	-
Proposed Model	Adaptive IDS	**99.51**	**91.48**	87.70	84.69	**99.23**	**0.25**

4.6Complexity analysis

The complexity of the proposed model determines its efficiency in detecting intrusions. The training of several classifiers, especially the C5.0 classifier, dominates Algorithm 2, which represents the RBE model. The total time complexity for the training phase is approximately $O(n \times m \times log(m))$, where m and n are the number of features and instances, respectively. On the other hand, the time complexity for the testing phase involves $O(n \times m)$ for comparing the data with stored rules and $O(m)$ for predictions using majority voting, resulting in a total of $O(n \times m)$. For Algorithm 3 on IF, the training phase takes $O(m \times f(n,d))$ with $O(1)$ for initializing the empty forest, $O(n)$ for selecting n subsamples, and $O(f(n,d))$ for constructing the tree with depth d and n data points. The testing phase takes $O(k \times m)$, with $O(k)$ for the number of instances, $O(m)$ for the number of trees, and $O(m)$ for calculating anomaly scores. For Algorithm 4 involving RF, the training phase has a time complexity of $O(n \times dlog(m))$ or $O(n \times m^2)$, including $O(1)$ for initializing an empty RF model. The testing phase takes $O(k \times m)$, with $O(k)$ for the number of instances and $O(m)$ for the number of trees. Moreover, the training phase of a model requires high CPU usage and memory due to the need to train multiple classifiers and store DTs and rulesets. During testing, CPU usage decreases but remains significant for real-time data comparison and prediction generation, while memory usage remains moderate due to stored rules and trees. Thus, the real-time processing capabilities are enhanced by the model's design, which distributes computational tasks across edge or fog nodes, thereby minimizing latency and ensuring efficient resource utilization.

4.7Security analysis

Assessing the effectiveness of the proposed model in identifying and mitigating various cybersecurity threats, as well as its resilience to different types of attacks and vulnerabilities, is crucial. The model's architecture, which combines signature-based, anomaly-based, and ensemble-based detection methods, is designed to provide comprehensive protection against a wide range of attack scenarios.

- *Known Attacks:* Signature-based detection excels at identifying well-documented attack patterns, such as malware signatures, known exploits, and specific vulnerability attacks. This method ensures that established threats are promptly detected and mitigated, reducing the risk of damage from known attack vectors.
- *Unknown Attacks:* Anomaly-based detection is critical for identifying novel or previously unseen threats by analysing deviations from normal behaviour. This includes zero-day attacks, polymorphic malware, and sophisticated phishing schemes that do not match known signatures. The adaptive nature of this method allows it to catch evolving threats that may bypass traditional signature-based systems.
- *Hybrid Attacks:* The ensemble-based approach integrates multiple detection techniques to handle complex and blended attack strategies that may use both known and unknown elements. For instance, an attack combining social engineering with sophisticated malware can be detected through the collaborative analysis of different models within the ensemble.
- *Evasion Attacks:* The RF algorithm's robustness and ability to handle high-dimensional and complex data improve the model's capacity to detect sophisticated evasion tactics. Ensemble learning enhances the system's capability to recognize obfuscated patterns and novel evasion methods that single classifiers might miss.
- *Spoofing and Replay Attacks:* The model's integration of multiple detection methods helps differentiate between legitimate and malicious activities, reducing the likelihood of successful spoofing and replay attacks.
- *DDoS Attacks:* The adaptive IDS model leverages its ensemble learning approach to identify abnormal traffic patterns and distinguish between legitimate traffic spikes and malicious DDoS attempts. This proactive analysis aids in mitigating the impact of such attacks.
- *Insider Threats:* Anomaly-based detection capabilities help identify suspicious behaviour or deviations from established patterns of authorized users, thereby reducing the risk of insider threats.
- *Dynamic Adaptation and Learning:* The model's ability to dynamically update decision boundaries and detection rules based on feedback from signature-based and anomaly-based methods enhances its adaptability to new and emerging threats. This continuous learning process refines the model's performance, ensuring it remains effective against evolving attack techniques and strategies.

By incorporating multiple detection methods, the model aims to minimize FPs and FNs. The ensemble approach allows for a more nuanced and accurate classification of threats, reducing the likelihood of legitimate activities being flagged as malicious and vice versa. This balanced decision-making process ensures a more reliable detection system, enhancing

overall security and reducing the risk of failing to detect real security threats or attacks.

4.8Considerations and challenges in system implementation

The adaptive IDS model must be implemented to support several security protocols, including secure sockets layer/transport layer security (SSL/TLS) for encrypted traffic, ensuring that the detection mechanisms can operate effectively even in encrypted environments. By decrypting traffic at the edge or fog nodes where IDS agents are deployed, the model maintains visibility into potential threats without compromising the security of the data in transit. However, a significant challenge lies in addressing real-time data streaming, buffering, and latency issues. The IDS must be designed to efficiently handle continuous data streams and buffer data without introducing unacceptable delays. This requires integrating advanced stream processing techniques and optimizing data buffering mechanisms to manage the high volume of network traffic in real time. To meet this challenge, the model must be designed to handle encrypted traffic by leveraging advanced decryption techniques at the edge or fog nodes. This approach ensures that the IDS can inspect the content of encrypted packets, identifying malicious activities hidden within encrypted traffic. Additionally, the model should employ techniques such as deep packet inspection (DPI) and SSL/TLS decryption to analyze encrypted traffic without compromising data privacy and security.

Distributing computational load at edge or fog nodes improves real-time processing capabilities while decreasing reliance on centralized resources. This distributed approach ensures the model can handle large amounts of network traffic without crashing. Optimizing the IDS for parallel processing and using GPUs may further enhance efficiency. Additionally, the use of modular design allows for deployment across various cloud environments, including edge and fog nodes, offering flexible and scalable integration that enhances security and threat management. Furthermore, the proposed IDS must integrate seamlessly with existing security infrastructures, such as firewalls and endpoint protection platforms, through standardized APIs and protocols. This integration allows for real-time data sharing and coordinated responses to detected threats. The IDS must trigger alerts and automated responses within the broader security ecosystem, enhancing the overall security posture. Also, the IDS framework must include mechanisms to handle system failures, data corruption, and network disruptions. Redundancy and failover strategies ensure continuous protection, with IDS agents at the edge or fog nodes operating independently during failures. Data validation and integrity checks prevent corruption, while backup sources ensure reliable analysis. Local caching at edge nodes enables real-time monitoring during network disruptions, synchronizing data once connectivity is restored. These measures strengthen IDS resilience for robust performance and security in adverse conditions.

5.Discussion

An adaptive IDS framework has been proposed to secure cloud computing environments against evolving cyber threats. This is achieved by deploying IDS agents at edge or fog nodes, thereby enhancing the real-time monitoring of network traffic and device behaviour. By performing data analysis locally, this approach reduces dependency on centralized resources, minimizes network latency, and improves overall system performance. The framework's multi-layered defence strategy, consisting of signature-based, anomaly-based, and ensemble learning techniques, ensures comprehensive security. Each layer focuses on different aspects of intrusion detection: 1) Signature-based detection utilizes RBE models, including RIPPER, C5.0, and PART classifiers, with ensemble learning; 2) Anomaly-based detection employs IF to detect deviations from normal behaviour; 3) Ensemble learning with RF combines the outputs from both signature-based and anomaly-based methods, enhancing detection accuracy and robustness against various types of attacks.

The methodology involved preprocessing and analysing large datasets such as UNSW-NB15, which contains 82,332 instances, and NSL-KDD, which has a total of 25,192 instances, providing a robust foundation for a comprehensive evaluation of the proposed model. The performance analysis with the UNSW-NB15 dataset for the proposed RBE, the first layer of defence, was carried out and compared with the other models (RIPPER, C5.0, and PART) used in the ensemble. The study reveals that although RIPPER has a better FAR (1.71%), PART has higher accuracy (86.49%) but lower attack accuracy (50.81%), while C5.0 has better average accuracy, attack accuracy, and ADR. The proposed RBE outperforms other models with improved performance metrics, including an accuracy of 89.70%, an average accuracy of 98.73%, an attack

accuracy of 65.49%, an ADR of 83.04%, and a FAR of 2.14%. The second layer of defence, the anomaly-based intrusion detection model using IF, is evaluated using the UNSW-NB15 dataset by replacing all 9 attack classes with 'attack' to simplify analysis. The model's performance on UNSW-NB15 test data showed improved precision (97.20%) for the attack class, better recall (97.17%) for the normal class, and better F-measure for both normal and attack classes. Additionally, the IF classifier used in this layer outperforms NB and OC-SVM in detecting anomaly-based intrusions, with a higher FAR of 6.93%. While OC-SVM has lower performance, IF has an average attack accuracy of 88.69% and an ADR of 80.22%.

Finally, with all three layers, the proposed adaptive IDS model with RF in the third layer is assessed using the UNSW-NB15 dataset. The results show strong performance in detecting normal traffic (precision: 97.66%, recall: 98.47%, F-measure: 98.06%) and generic traffic (precision: 98.52%, recall: 98.63%, F-measure: 98.57%), but lower performance in detecting Shellcode (precision: 58.68%), lower recall (71.83%) and F-measure (58.59%) for analysis, and lower recall (50%) for worms. The performance of the proposed adaptive IDS with RF was compared with other models such as MNC, AB, XGB, SVM, MLP, and CART. Results show that MNC and SVM classifiers have lower performance, while AB and CART have better accuracy but lower average and attack accuracy. The proposed IDS with RBE, IF, and RF algorithms have higher accuracy (92.68%) and a lower FAR (1.53%). With the NSL-KDD dataset, the model performed best with normal, DoS, and Probes classes but struggled with R2L and U2R attacks. Despite high precision, it had lower recall for R2L and U2R attacks, leading to high FNs. Furthermore, the performance metrics of the proposed adaptive IDS were also compared to those of other models, including RIPPER, C5.0, PART, RBE, and IF. The C5.0 classifier has higher average accuracy and attack accuracy, while the PART and IF models have lower average and attack accuracy. The proposed adaptive IDS has higher accuracy (99.51%) and ADR (99.23%).

Thus, with the UNSW-NB15 dataset, the proposed method performs well with normal and generic classes due to their higher prevalence, providing more training data. However, it struggles with shellcode, analysis, and DoS categories because of class imbalance and complexity. The NSL-KDD dataset shows that the method excels in common attacks (Normal, DoS, Probes) but struggles with less frequent types (R2L, U2R). Despite these challenges, the proposed adaptive IDS outperforms state-of-the-art models in accuracy (92.68%), attack accuracy, and MFM. It surpasses unified IDS, edge-based HIDS, and DENDRON, which has an average accuracy of 90.54% and an attack accuracy of 88.44%. The model's highest ADR of 99.23% and lowest FAR of 0.25% highlights its superiority. By combining signature-based, anomaly-based, and ensemble-based methods, the RF algorithm enhances detection accuracy and resilience. The model can update decision boundaries based on feedback, adapting to evolving threats and reducing vulnerabilities in static detection models.

Furthermore, to enhance the performance of the IDS, organizations should explore integrating additional advanced ML techniques and investigate the interplay between various detection layers to assess their combined impact on accuracy and robustness. Advanced models for class imbalance and attack complexity are essential for threat detection while exploring the adaptability of the IDS framework in diverse cloud environments and dynamic threat landscapes enhances understanding. Practically, deploying the adaptive IDS framework in real-world cloud environments is essential for validating its effectiveness. Continuous updates based on feedback mechanisms should be implemented to ensure ongoing adaptability to evolving threats. Organizations must conduct extensive testing under various network conditions and attack scenarios to ensure reliability. Implementing comprehensive adversarial testing and integrating defensive mechanisms will bolster the model's resilience. Furthermore, optimizing the model for processing speed and memory utilization is crucial for real-time applications. Regular cross-validation with diverse datasets will help maintain the IDS's performance across different environments and attack vectors.

5.1Study implications

The study contributes to the theoretical understanding of IDSs by integrating signature-based, anomaly-based, and ensemble learning techniques into a unified adaptive framework. It advances the knowledge of IDS architectures by demonstrating how multi-layered defence strategies can enhance detection accuracy and robustness against evolving cyber threats. The research also sheds light on the efficacy of combining rule-based models, such as RIPPER, C5.0, and PART, with anomaly detection methods like IF. The theoretical implications

underscore the importance of adaptive mechanisms in IDS, emphasizing their role in maintaining security in dynamic cloud computing environments.

Practically, the proposed adaptive IDS framework offers a robust solution for securing cloud computing environments against diverse and evolving cyber threats. Deploying IDS agents at edge or fog nodes enhances real-time monitoring, reduces dependency on centralized resources, and minimizes network latency. The framework's ability to detect a wide range of attacks with high accuracy and low FP rates makes it suitable for deployment in real-world scenarios. Additionally, its adaptability to new threats and integration of feedback mechanisms ensures ongoing security and resilience, providing a reliable defence mechanism for organizations seeking to protect their cloud infrastructure.

5.2Limitations of the study

Though the proposed model demonstrates enhanced security in EC and FC environments, several limitations need to be addressed. First, the model's performance is primarily demonstrated using the UNSW-NB15 and NSL-KDD datasets, which may not cover all possible real-world attack scenarios. While these datasets are comprehensive, the inclusion of an additional synthetic dataset could further validate the effectiveness of the technique. Second, while CRS balances training data representation, the dataset imbalance may impact model performance and generalization, affecting IDS effectiveness in real-world scenarios with varying class distributions.

Third, the current study lacks detailed information on the computational cost and resource utilization of the adaptive IDS model. Future research should focus on improving the model's efficiency in terms of CPU usage, memory requirements, and real-time processing capabilities, especially for EC deployment. Fourth, the effectiveness of the model in real-world scenarios remains unverified. To confirm its practical utility, the model's performance under real network conditions and traffic loads should be thoroughly tested in real cloud and edge environments for practical applicability and reliability. Fifth, the evaluation lacks comprehensive testing for model resilience against adversarial attacks, highlighting the need for comprehensive testing to ensure performance and security under malicious conditions. Additionally, the model's robustness against varying network conditions remains unverified and requires further evaluation. Also, the model's generalizability across diverse network environments and attack scenarios necessitates further investigation and cross-validation with diverse datasets to assess its effectiveness in real-world scenarios. Furthermore, the model's scalability needs examination to ensure it can handle large volumes of network traffic data without performance degradation. Finally, to ensure effective deployment in real-time applications, the model's performance parameters, including processing speed and memory utilization, must be evaluated to confirm its efficiency and ability to meet the demands of real-time systems.

A complete list of abbreviations is listed in *Appendix I.*

6.Conclusion and future work

The rising intelligence of cyber threats and the rapid growth of cloud computing make it critical to develop reliable and dynamic IDS for protecting the cloud environment. An adaptive intrusion detection framework has been proposed for cloud computing environments to overcome the limitations of traditional detection methods. Implementing the proposed framework at the EC and FC nodes provides real-time threat detection and response. A multi-layered defence strategy that integrates the RBE model and IF classifier helps to identify known and unknown attacks, respectively. Furthermore, the additional layer employing ensemble learning with these two layers demonstrates superior performance compared to conventional methods. The performance of the proposed model is evaluated with various traditional models using two publicly available network traffic datasets: the UNSW-NB15 and NSL-KDD datasets. The model is evaluated using various performance metrics to showcase its effectiveness. The model shows better results with 92.68% accuracy, an 88.03% ADR, and a low FAR of 1.53% with the UNSW-NB15 dataset. Upon evaluation with the NSL-KDD dataset, the proposed model shows better accuracy (99.51%), an MFM (91.48%), an ADR of 99.23%, and a lower FAR of 0.25%. Based on the findings, the suggested framework is superior to both traditional and other recent models such as DENDRON, stacking ensemble, unified IDS, integrated rule-based detection model, and HIDS in identifying security breaches. The results show that the adaptive IDS is effective in protecting the cloud from constantly evolving cyber threats by making it more resilient.

Even though the performance of the proposed model is promising, several issues need to be addressed in

the future. First, future work should enhance algorithm efficiency and accuracy by utilizing advanced ML and DL techniques, refining anomaly detection algorithms, and enhancing ensemble learning methods to further improve detection accuracy and reduce FPs. While CRS helps balance the training set, the inherent class imbalance in the original dataset may still impact the model's performance and generalization. Future work should address this by exploring advanced resampling techniques and cost-sensitive learning. To assess the scalability and adaptability of the framework, it should be deployed and tested in a real-time environment with large network traffic data. This will help evaluate how well the model performs under dynamic conditions and varied traffic loads. Additionally, the model must be further enhanced to detect advanced persistent threats and evaluate its resilience against adversarial attacks. The robustness of the model against varying network conditions needs further evaluation. Comprehensive adversarial testing under diverse network scenarios is essential to assess the model's robustness against input manipulation, model poisoning, and other malicious attacks. The performance of the model must be evaluated with other parameters, such as processing speed and memory utilization, to ensure it meets the demands of real-time systems, particularly in EC environments. While the model currently employs ensemble learning methods for final decision-making, incorporating online learning techniques could allow for dynamic behaviour analysis and adaptation to new threats in real time. Further research should focus on improving the model's generalization across different network environments and attack scenarios through cross-validation with diverse datasets. This will help determine its effectiveness beyond the training data and in varied real-world situations.

Acknowledgment
None.

Conflicts of interest
The authors have no conflicts of interest to declare.

Data availability
The data used in this study were obtained from publicly accessible repositories: (1) UNSW NB-15 testing ser, available at https://github.com/Nir-J/ML-Projects/blob/master/UNSW-Network_Packet_Classification/UNSW_NB15_testing-set.csv; (2) NSL-KDD Train 20percent dataset, available at https://www.kaggle.com/datasets/hassan06/nslkdd/data?select=KDDTrain%2B_20Percent.txt.

Author's contribution statement
K. Vani: Conceptualization and design of the work, literature review, data collection, implementation, experimental analysis and interpretation of results, and manuscript preparation. **S. P. Swornambiga:** Conceptualization and design of the work, supervision, review of the results and critical assessment of the manuscript.

References
[1] Khater BS, Abdul WAW, Idris MY, Hussain MA, Ibrahim AA, Amin MA, et al. Classifier performance evaluation for lightweight IDS using fog computing in IoT security. Electronics. 2021; 10(14):1-52.

[2] Yu W, Liang F, He X, Hatcher WG, Lu C, Lin J, et al. A survey on the edge computing for the internet of things. IEEE Access. 2017; 6:6900-19.

[3] Alzubi OA, Qiqieh I, Alzubi JA. Fusion of deep learning based cyberattack detection and classification model for intelligent systems. Cluster Computing. 2023; 26(2):1363-74.

[4] El-sayed H, Sankar S, Prasad M, Puthal D, Gupta A, Mohanty M, et al. Edge of things: the big picture on the integration of edge, IoT and the cloud in a distributed computing environment. IEEE Access. 2017; 6:1706-17.

[5] Alzubi OA, Alzubi JA, Alazab M, Alrabea A, Awajan A, Qiqieh I. Optimized machine learning-based intrusion detection system for fog and edge computing environment. Electronics. 2022; 11(19):1-16.

[6] Zwayed FA, Anbar M, Sanjalawe Y, Manickam S. Intrusion detection systems in fog computing–a review. In third international conference on advances in cyber security 2021 (pp. 481-504). Springer Singapore.

[7] Aliyu F, Sheltami T, Shakshuki EM. A detection and prevention technique for man in the middle attack in fog computing. Procedia Computer Science. 2018; 141:24-31.

[8] Gao J, Chai S, Zhang B, Xia Y. Research on network intrusion detection based on incremental extreme learning machine and adaptive principal component analysis. Energies. 2019; 12(7):1-17.

[9] Sousa BF, Abdelouahab Z, Lopes DC, Soeiro NC, Ribeiro WF. An intrusion detection system for denial of service attack detection in internet of things. In proceedings of the second international conference on internet of things, data and cloud computing 2017 (pp. 1-8). ACM.

[10] Muhammad LI, Zambuk FU, Muhammad D, Shittu F, Gital AY, Ibrahim KM, et al. An improved security framework for cyber malicious device detection in fog environments. International Journal of Engineering Research & Technology. 2023; 12(2):190-6.

[11] Saravanan A, Bama SS. CloudSec (3FA): a multifactor with dynamic click colour-based dynamic authentication for securing cloud environment. International Journal of Information and Computer Security. 2023; 20(3-4):269-94.

[12] Arumugam S. An effective hybrid encryption model using biometric key for ensuring data security. International Arab Journal of Information Technology. 2023; 20(5):796-807.
[13] Roopak M, Tian GY, Chambers J. An intrusion detection system against DDoS attacks in IoT networks. In 10th annual computing and communication workshop and conference 2020 (pp. 562-7). IEEE.
[14] Singh A, Chatterjee K, Satapathy SC. An edge based hybrid intrusion detection framework for mobile edge computing. Complex & Intelligent Systems. 2022; 8(5):3719-46.
[15] Aldaej A, Ahanger TA, Ullah I. Deep learning-inspired IoT-IDS mechanism for edge computing environments. Sensors. 2023; 23(24):1-20.
[16] Raponi S, Caprolu M, Di PR. Intrusion detection at the network edge: solutions, limitations, and future directions. In third international conference on edge computing 2019 (pp. 59-75). Springer International Publishing.
[17] Halimaa A, Sundarakantham K. Machine learning based intrusion detection system. In 3rd international conference on trends in electronics and informatics 2019 (pp. 916-20). IEEE.
[18] Khraisat A, Gondal I, Vamplew P, Kamruzzaman J, Alazab A. Hybrid intrusion detection system based on the stacking ensemble of C5 decision tree classifier and one class support vector machine. Electronics. 2020; 9(1):1-18.
[19] Judith A, Kathrine GJ, Silas S. Efficient deep learning-based cyber-attack detection for internet of medical things devices. Engineering Proceedings. 2023; 59(1):1-10.
[20] Alzubi JA, Alzubi OA, Qiqieh I, Singh A. A blended deep learning intrusion detection framework for consumable edge-centric IOMT industry. IEEE Transactions on Consumer Electronics. 2024; 70(1):2049-57.
[21] Adejimi AO, Sodiya AS, Ojesanmi OA, Falana OJ, Tinubu CO. A dynamic intrusion detection system for critical information infrastructure. Scientific African. 2023; 21(2023):1-12.
[22] Kumar V, Kumar V, Singh N, Kumar R. Enhancing intrusion detection system performance to detect attacks on edge of things. SN Computer Science. 2023; 4(6):802.
[23] Kumar P, Gupta GP, Tripathi R. A distributed ensemble design based intrusion detection system using fog computing to protect the internet of things networks. Journal of ambient intelligence and humanized Computing. 2021; 12(10):9555-72.
[24] Abualghanam O, Alazzam H, Alhenawi EA, Qatawneh M, Adwan O. Fusion-based anomaly detection system using modified isolation forest for internet of things. Journal of Ambient Intelligence and Humanized Computing. 2023; 14(1):131-45.
[25] Bakro M, Kumar RR, Alabrah AA, Ashraf Z, Bisoy SK, Parveen N, et al. Efficient intrusion detection system in the cloud using fusion feature selection approaches and an ensemble classifier. Electronics. 2023; 12(11):1-27.
[26] Alsubhi K. A secured intrusion detection system for mobile edge computing. Applied Sciences. 2024; 14(4):1-14.
[27] Sajid J, Hayawi K, Malik AW, Anwar Z, Trabelsi Z. A fog computing framework for intrusion detection of energy-based attacks on UAV-assisted smart farming. Applied Sciences. 2023; 13(6):1-23.
[28] Lesouple J, Baudoin C, Spigai M, Tourneret JY. Generalized isolation forest for anomaly detection. Pattern Recognition Letters. 2021; 149:109-19.
[29] Yao H, Gao P, Zhang P, Wang J, Jiang C, Lu L. Hybrid intrusion detection system for edge-based IIoT relying on machine-learning-aided detection. IEEE Network. 2019; 33(5):75-81.
[30] Gaikwad DP. Intrusion detection system using ensemble of rule learners and first search algorithm as feature selectors. International Journal of Computer Network and Information Security. 2021; 13(4):26-34.
[31] Kalaivani K, Chinnadurai M. A hybrid deep learning intrusion detection model for fog computing environment. Intelligent Automation & Soft Computing. 2021; 30(1):1-15.
[32] Aslan Ö, Aktuğ SS, Ozkan-okay M, Yilmaz AA, Akin E. A comprehensive review of cyber security vulnerabilities, threats, attacks, and solutions. Electronics. 2023; 12(6):1-42.
[33] Pinto A, Herrera LC, Donoso Y, Gutierrez JA. Survey on intrusion detection systems based on machine learning techniques for the protection of critical infrastructure. Sensors. 2023; 23(5):1-18.
[34] Bama SS, Ahmed MI, Saravanan A. Network intrusion detection using clustering: a data mining approach. International Journal of Computer Applications. 2011; 30(4):14-7.
[35] Panda M, Abraham A, Patra MR. Discriminative multinomial naive bayes for network intrusion detection. In sixth international conference on information assurance and security 2010 (pp. 5-10). IEEE.
[36] Tao X, Peng Y, Zhao F, Zhao P, Wang Y. A parallel algorithm for network traffic anomaly detection based on isolation forest. International Journal of Distributed Sensor Networks. 2018; 14(11):1-11.
[37] Ramkumar MP, Daniya T, Paul PM, Rajakumar S. Intrusion detection using optimized ensemble classification in fog computing paradigm. Knowledge-Based Systems. 2022; 252:109364.
[38] Almogren AS. Intrusion detection in edge-of-things computing. Journal of Parallel and Distributed Computing. 2020; 137:259-65.
[39] Pacheco J, Benitez VH, Felix-herran LC, Satam P. Artificial neural networks-based intrusion detection system for internet of things fog nodes. IEEE Access. 2020; 8:73907-18.
[40] Sahar N, Mishra R, Kalam S. Deep learning approach-based network intrusion detection system for fog-assisted IoT. In proceedings of international

conference on big data, machine learning and their applications 2021 (pp. 39-50). Springer Singapore.
[41] Papamartzivanos D, Mármol FG, Kambourakis G. Dendron: genetic trees driven rule induction for network intrusion detection systems. Future Generation Computer Systems. 2018; 79:558-74.
[42] Moustafa N, Slay J. The evaluation of network anomaly detection systems: statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. Information Security Journal: A Global Perspective. 2016; 25(1-3):18-31.
[43] Kumar V, Sinha D, Das AK, Pandey SC, Goswami RT. An integrated rule based intrusion detection system: analysis on UNSW-NB15 data set and the real time online dataset. Cluster Computing. 2020; 23:1397-418.
[44] Sohal AS, Sandhu R, Sood SK, Chang V. A cybersecurity framework to identify malicious edge device in fog computing and cloud-of-things environments. Computers & Security. 2018; 74:340-54.
[45] Shitharth S, Mohammed GB, Ramasamy J, Srivel R. Intelligent intrusion detection algorithm based on multi-attack for edge-assisted internet of things. In security and risk analysis for intelligent edge computing 2023 (pp. 119-35). Cham: Springer International Publishing.
[46] Liang C, Shanmugam B, Azam S, Karim A, Islam A, Zamani M, et al. Intrusion detection system for the internet of things based on blockchain and multi-agent systems. Electronics. 2020; 9(7):1-27.
[47] Sudqi KB, Abdul WAW, Idris MY, Abdulla HM, Ahmed IA. A lightweight perceptron-based intrusion detection system for fog computing. Applied Sciences. 2019; 9(1):1-21.
[48] Syed NF, Ge M, Baig Z. Fog-cloud based intrusion detection system using recurrent neural networks and feature selection for IoT networks. Computer Networks. 2023; 225:109662.
[49] Kumar R, Kumar P, Tripathi R, Gupta GP, Garg S, Hassan MM. A distributed intrusion detection system to detect DDoS attacks in blockchain-enabled IoT network. Journal of Parallel and Distributed Computing. 2022; 164:55-68.
[50] Hao WT, Lu Y, Dong RH, Shui YL, Zhang QY. Adaptive intrusion detection model based on CNN and C5.0 classifier. International Journal of Network Security. 2022; 24(4):648-60.
[51] Kumar V, Das AK, Sinha D. UIDS: a unified intrusion detection system for IoT environment. Evolutionary Intelligence. 2021; 14(1):47-59.
[52] Saravanan A, Bama SS, Kadry S, Ramasamy LK. A new framework to alleviate DDoS vulnerabilities in cloud computing. International Journal of Electrical & Computer Engineering. 2019; 9(5):4163-75.
[53] Potdar K, Pardawala TS, Pai CD. A comparative study of categorical variable encoding techniques for neural network classifiers. International Journal of Computer Applications. 2017; 175(4):7-9.
[54] Bama SS, Saravanan A. Efficient classification using average weighted pattern score with attribute rank based feature selection. International Journal of Intelligent Systems and Applications. 2019; 11(7):29-42.
[55] Urbanowicz RJ, Meeker M, La CW, Olson RS, Moore JH. Relief-based feature selection: introduction and review. Journal of Biomedical Informatics. 2018; 85:189-203.
[56] Permadi VA, Tahalea SP, Agusdin RP. K-means and elbow method for cluster analysis of elementary school data. Progres Pendidikan. 2023; 4(1):50-7.
[57] Moustafa N, Slay J. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In military communications and information systems conference (MilCIS) 2015 (pp. 1-6). IEEE.
[58] Tavallaee M, Bagheri E, Lu W, Ghorbani AA. A detailed analysis of the KDD CUP 99 data set. In symposium on computational intelligence for security and defense applications 2009 (pp. 1-6). IEEE.
[59] Solekha NA. Analysis of NSL-KDD dataset for classification of attacks based on intrusion detection system using binary logistics and multinomial logistics. In seminar nasional official statistics 2022 (pp. 507-20).
[60] Saravanan A, Bama SS. Multi-model anti-Ddos framework for detection and mitigation of high rate Ddos attacks in the cloud environment. International Journal of Scientific & Technology Research. 2020; 9(3):4503-11.

K. Vani is currently working as an Assistant Professor in the Department of Computing - Decision and Computing Sciences at Coimbatore Institute of Technology, Coimbatore. She has 7 years of teaching experience. Her research areas include Big Data Analytics and Cloud Security in IoT. She has participated in two international conferences and presented four research papers.
Email: vanithanu@gmail.com

Dr. S. P. Swornambiga is currently working as an Associate Professor in the Department of Computer Applications, CMS College of Science & Commerce, Coimbatore with a teaching experience of 22 years. Her research area Includes Networks, Network Security, Web Mining and Image Processing. She has attended 2 International Conferences. She has published 20 journals in Scopus and international referred journals. She has organized many programmes. She has guided 1 M.Phil and 6 Ph.D scholars.
Email: swornagoms@gmail.com

Appendix I

S. No.	Abbreviation	Description
1	AB	Adaptive Boosting
2	ADR	Attack Detection Rate
3	CART	Classification and Regression Trees
4	CNN	Convolutional Neural Network
5	CPU	Central Processing Unit
6	CRS	Cluster-Based Representative Sampling
7	CSA	Crow Search Algorithm
8	CSV	Comma-Separated Values
9	DBN	Deep Belief Network
10	DDoS	Distributed Denial of Service
11	DENDRON	Derivation of Efficient Network Detection Rules by Optimized Neural methods
12	DL	Deep Learning
13	DoS	Denial of Service
14	DPI	Deep Packet Inspection
15	DT	Decision Tree
16	EC	Edge Computing
17	EHIDF	Edge-based Hybrid Intrusion Detection Framework
18	EIF	Extended Isolation Forest
19	EoT	Edge-of-Things
20	FAR	False Alarm Rate
21	FC	Fog Computing
22	FN	False Negative
23	FP	False Positive
24	GIF	Generalised Isolation Forest
25	GPU	Graphical Processing Unit
26	HIDS	Hybrid IDS
27	ID3	Iterative Dichotomiser3
28	IDS	Intrusion Detection System
29	IF	Isolation Forest
30	IoT	Internet of Things
31	IP	Internet Protocol
32	KDD	Knowledge Discovery and Data Mining
33	LSTM	Long Short-Term Memory
34	MFM	Mean F-Measure
35	MIM	Man In the Middle
36	ML	Machine Learning
37	MLP	Multilayer Perceptron
38	MNB	Multinomial Naïve Bayes
39	MNC	Multinominal Classifier
40	NB	Naïve Bayes
41	NN	Neural Networks
42	NSL	National Security Laboratory
43	OC-SVM	One-Class Support Vector Machines
44	PART	Partial Decision Tree
45	PCA	Principal Component Analysis
46	R2L	Remote-to-Local
47	RBE	Rule-Based Ensemble
48	RF	Random Forest
49	RIPPER	Repeated Incremental Pruning to Produce Error Reduction
50	RNN	Recurrent Neural Network
51	SEC-IDS	Secured Edge Computing IDS
52	SMO	Sequential Minimal Optimization
53	SSL	Secure Sockets Layer
54	SVM	Support Vector Machine
55	TLS	Transport Layer Security
56	TN	True Negative
57	TP	True Positive
58	U2R	User-to-Root attacks
59	UAV	Unmanned Aerial Vehicles
60	UNSW-NB	University of New South Wales Network-Based
61	VHD	Virtual Honeypot Device
62	XGB	Extreme Gradient Boosting