Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.059805

ARTICLE

Intrumer: A Multi Module Distributed Explainable IDS/IPS for Securing Cloud Environment

Nazreen Banu A[*] **and S.K.B. Sangeetha**

Department of Computer Science and Engineering, SRM Institute of Science and Technology, Vadapalani Campus, Chennai, 600026, Tamil Nadu, India

*Corresponding Author: Nazreen Banu A. Email: nazreenbanu8@gmail.com

Received: 17 October 2024 Accepted: 05 December 2024 Published: 03 January 2025

ABSTRACT

The increasing use of cloud-based devices has reached the critical point of cybersecurity and unwanted network traffic. Cloud environments pose significant challenges in maintaining privacy and security. Global approaches, such as IDS, have been developed to tackle these issues. However, most conventional Intrusion Detection System (IDS) models struggle with unseen cyberattacks and complex high-dimensional data. In fact, this paper introduces the idea of a novel distributed explainable and heterogeneous transformer-based intrusion detection system, named INTRUMER, which offers balanced accuracy, reliability, and security in cloud settings by multiple modules working together within it. The traffic captured from cloud devices is first passed to the TC&TM module in which the Falcon Optimization Algorithm optimizes the feature selection process, and Naïve Bayes algorithm performs the classification of features. The selected features are classified further and are forwarded to the Heterogeneous Attention Transformer (HAT) module. In this module, the contextual interactions of the network traffic are taken into account to classify them as normal or malicious traffic. The classified results are further analyzed by the Explainable Prevention Module (XPM) to ensure trustworthiness by providing interpretable decisions. With the explanations from the classifier, emergency alarms are transmitted to nearby IDS modules, servers, and underlying cloud devices for the enhancement of preventive measures. Extensive experiments on benchmark IDS datasets CICIDS 2017, Honeypots, and NSL-KDD were conducted to demonstrate the efficiency of the INTRUMER model in detecting network traffic with high accuracy for different types. The proposed model outperforms state-of-the-art approaches, obtaining better performance metrics: 98.7% accuracy, 97.5% precision, 96.3% recall, and 97.8% F1-score. Such results validate the robustness and effectiveness of INTRUMER in securing diverse cloud environments against sophisticated cyber threats.

KEYWORDS

Cloud computing; intrusion detection system; transformers; and explainable artificial intelligence (XAI)

1 Introduction

The Internet of Things (IoT) has been the transformative force in modern technology, leading to breakthroughs in many different environments and applications. As the systems of IoT advance, so does the potential of these systems to change the course of society and open up new possibilities that have garnered attention from researchers and industry leaders [1,2]. IoT devices are now inseparable parts of most sectors with new applications constantly springing up [3]. This has been achieved from decades of scientific development where challenges thought to be daunting in tasks have now improved the productivity and efficiency [4]. However, the fast development of IoT devices faces serious challenges especially on issues of storage and security [5,6]. As a way of overcoming storage shortages, cloud computing has come as an added technology. Cloud computing delivers IT services, platforms, and software over the internet, offering scalability, accessibility, and reliability. Often called "Utility Computing," it is widely deployed in private, public, and hybrid forms. When integrated with IoT, the resulting Cloud-IoT paradigm enables dependability and scalability, providing an optimal environment for IoT deployments [7]. However, many IoT applications require computational capabilities, low latency, mobility support, and robust distributed systems—requirements that are not fully satisfied by traditional cloud computing frameworks [8]. This indicates the necessity of innovative technologies to complement the existing Cloud-IoT architecture. Security is a major concern in Cloud-IoT environments as the number of security breaches has increased significantly, especially in the virtual network layer [9]. Though numerous IDS are present in the cloud, traditional IDS is still of no use to capture such dynamically high-traffic natures of the cloud environments [10–12]. The basic nature of virtualization and internet protocol usage in cloud infrastructure is itself a vulnerable environment to attack for various kinds of attack modes, including zero-day attack [13]. For unfamiliar threats, such amounts of data flow within a cloud environment provide a challenging environment for most organizations. Traditional methods are generally ineffective for the identification and prevention of these attacks efficiently [14]. Machine Learning (ML) has recently emerged as a promising answer to enable the detection of both classic and zero-day attacks. ML relies on algorithms capable of finding patterns in data to make predictions. It therefore combines computer science and statistical methods to enhance the predictive capacity. The ML technique is generally classified into supervised, unsupervised, and semi-supervised learning [15,16]. Supervised learning is based on labeled data for training classification models, whereas unsupervised learning finds hidden patterns without specific guidance [17]. Algorithms like KNN, Naïve Bayes (NB), Decision Trees (DT), Logistic Regression (LR), and Support Vector Machines (SVM) are very commonly used. For example, K-means clustering is a very popular unsupervised learning algorithm [18]. Moreover, Deep Learning (DL) allows sophisticated data representations, bringing about significant advancements in all domains [19]. It develops a novel IDS for the Cloud-IoT environment termed as INTRUMER model. This proposes addressing the existing limitations by bringing advanced modules together. In short, the contributions of the proposed study are:

Multi-module heterogeneous transformer architecture: This is our first work adopting this methodology with a feature collaboration layer that enhances detection accuracy but limits complexity in IDS for Cloud-IoT environments.

Optimized feature selection and classification: The Falcon Optimization Algorithm (FOA) is used for feature selection, and the Naïve Bayes (NB) algorithm is used for efficient classification of dense scalar and categorical features, which improves the interpretation of real-time IoT network traffic.

Explainable Prevention Module (XPM): This module produces interpretable emergency alarms for IoT devices and servers, thereby enhancing the trustworthiness and proactive prevention capabilities of IDS.

2 Literature Survey

Wu et al. [20] introduced transformer entrenched IDS framework. Leveraging the power of Transformer architectures, the authors demonstrate a commendable enhancement in the accuracy of detection of breaches. The study not only addresses the challenge of handling diverse and dynamic cyber threats but also sheds light on the potential of Transformer models in cybersecurity applications. In dynamic cloud environments, traditional intrusion detection systems (IDS), which are mostly centralized and rule-based, sometimes struggle with scalability and adaptation. Distributed systems and machine learning models have been studied in more recent studies, although many of them are unable to justify their choices. Though their use in cloud security is still in its infancy, transformer-based techniques have demonstrated promise in interpreting complex data patterns in other sectors. INTRUMER distinguishes itself by fusing a transformer-based paradigm created especially for cloud environments with a distributed, explainable architecture. Ho et al. [21] designed the CNN entrenched IDS to effectively detect both known and innovative cyberattacks. The utilization of CNNs showcases a pragmatic approach to capturing spatial dependencies in network data. The paper contributes significantly to the domain by bridging the gap between traditional detection methods and the evolving landscape of cyber threats. In [22], IDS-INT introduces a Transformer-based transfer learning approach for addressing the challenges posed by imbalanced traffic on networks. The significance of transfer learning in building robust models is emphasized throughout the research, especially in situations where class inequalities are common. The research adds significant value to the field by improving the adaptability and generalization capabilities of intrusion detection systems. In [23], a real-time anomaly detection system with intelligence that could be used for cloud communications is proposed. The combination of neural network methods and temporal data summarization demonstrates a thorough approach to anomaly detection. The paper showcases the significance of real-time adaptability in cloud environments and contributes to the development of effective solutions for securing cloud-based communications. Elmasry et al. [24] focused on integration, this paper presents a design for an IDS system that collaborates with cloud services. The approach signifies the growing need for comprehensive security solutions that extend beyond individual systems. This research emphasis on collaboration and integration presents a forward-thinking perspective to address the multi-faceted challenges associated with cloud security.

Hierarchical transformers were used by Huang et al. [25] to identify irregularities in the Internet of Things environment. The hierarchical structure helps the model identify abnormalities more accurately, which allows it to understand both local and global patterns in log data. The authors' dedication to incorporating preexisting works is demonstrated by the usage of transformers. In an IoT setting, Ma et al. [26] used kernel SVM for IDS. The use of SVMs in combination with kernel techniques highlights the paper's emphasis on the data's non-linear connections. Using network traffic records, the paper provides a comprehensive evaluation of the proposed model, demonstrating its capacity to successfully identify aberrant patterns. The landscape of anomaly detection gains a useful dimension with the incorporation of kernel approaches. Nie et al. [27] concentrated on the green IoT and use an intrusion detection technique based on DDPG. This study's application of reinforcement learning techniques to the intrusion detection sector is one of its standout features. The writers take a proactive stance in addressing the issues of energy efficiency in IoT setups. The algorithm is a promising contribution to green IoT security, as demonstrated by the experimental results, which show that it can adapt and learn in dynamic contexts. A network intrusion detection technique that combines DTTSVM with hierarchical clustering is presented by Zou et al. [28]. By combining the advantages of support vector machines and decision trees, the hybrid approach improves IDS performance. The stability of the model is further improved by adding hierarchical clustering. The assertion that this approach is excellent at precisely identifying network intrusions is supported by the experimental validation. The study [29], which focuses on large data platforms, suggests an IDS method that uses an ensemble of SVM and the CGO algorithm. While the CGO method helps to optimize the SVM parameters, the ensemble technique seeks to improve robustness and model generalization. Through considerable experimentation, the study presents a convincing case for the effectiveness of the research. An important advancement in intrusion detection is the combination of metaheuristic optimization with ensemble learning.

A dual-stage IDS model that combines AEs and LSTM networks is presented by Mushtaq et al. [30]. The combination of these methods demonstrates a strong strategy for locating and reducing any security risks. The system's usefulness is demonstrated by the amalgamation of findings, which makes it a significant addition to the intrusion detection area. The goal of Krishnaveni et al. [31] is to apply ensemble techniques to improve classification and feature selection in NIDS in cloud computing environments. By selecting the most relevant features using ensemble-based models, this study's accuracy and efficiency are increased. This comprehensive experimental evaluation, which shows the efficacy of the proposed technique, contributes significantly to the field of intrusion detection on cloud platforms. Recurrent neural networks and metaheuristic feature selection algorithms are combined by Saisindhu Theja et al. [32] to provide a novel method for cloud computing DoS attack detection. Enhancing the system's capacity to adapt to evolving threats involves the addition of metaheuristic algorithms. Javadpour et al. [33] described a particular intrusion detection and prevention system designed for cloud-based IoT systems, utilizing a distributed multi-agent architecture. The innovative architecture of the system addresses the unique challenges posed by IoT devices in a cloud setting. A key step in securing IoT ecosystems in cloud environments, the thorough evaluation of the performance and scalability of the proposed system indicates its potential usefulness.

Okey et al. [34] examined the application of transfer learning in IDS for Cloud-IoT devices using an OCNN. The use of transfer learning enhances the model's flexibility in different IoT device scenarios. This study advances the dynamic field of intrusion detection in the context of Cloud-IoT. Selecting characteristics for classifiers utilized in CIoT for intrusion detection, Sangaiah et al. [35] presented a hybrid technique that combines heuristics. Artificial intelligence improves the system's efficiency and makes it possible to identify possible threats with greater accuracy. The study examines how heuristics and AI can work together, offering important new information for creating reliable IDS for cloud systems. To address the critical issue of interpretability in machine learning models, Gaitan-Cardenas et al. [36] suggested XML-entrenched IDS for CIoT. By increasing the models' transparency, this study not only finds intrusions but also offers lucid insights into the decision-making process, which promotes greater comprehension and confidence in the security measures used. An IDS framework that functions in multi-cloud and IoT contexts was developed by Nizamuddin et al. [37]. Innovative integration of intricate algorithms for improved threat identification is demonstrated by the use of a swarm-entrenched DL classifier. The study highlights how flexible the suggested architecture is in a variety of cloud and IoT environments. GOSVM is designed by Arunkumar et al. [38] to identify malicious attacks in cloud environments. The research enhances SVM performance by utilizing GO. Accuracy and efficiency are shown to be promising when optimization approaches and SVM are combined. By combining the HASH and LSTM approaches, Ali et al. [39] presented a unique IDS for the IoT-Cloud platform. The study also discusses data encryption and user authentication techniques, highlighting a thorough approach to security in IoT-Cloud settings. Improvements in detection precision and overall system security are demonstrated by the suggested system.

3 Intrumer Framework

A distributed, explainable, and heterogeneous transformer-based intrusion detection system designed especially for cloud environments is called INTRUMER. It overcomes the drawbacks of conventional IDS, which frequently have trouble processing data in real time, adapting to changing traffic and scaling in cloud-based systems. Additionally, conventional IDS models usually make decisions that are hard to explain, which makes it challenging to understand alerts that occur. The INTRUMER model is a sophisticated intrusion detection system (IDS) that was created in order to get around some of the drawbacks of traditional IDS, namely their dependence on static rule sets and lack of flexibility. To improve detection capabilities and reduce false positives, it combines advanced algorithms with machine learning approaches. The proposed INTRUMER model integrates distributed, explainable, and heterogeneous transformer-based techniques to set itself apart from other IDS models. INTRUMER includes a multi-layered approach that make use of the Heterogeneous Attention Transformer (HAT) module for adaptive attention to a variety of feature types and contextual information. In contrast to traditional IDS models that usually rely on either homogeneous architectures or single-stage detection processes. Furthermore, unlike existing models that lack these features, INTRUMER's explainable components such as the Naïve Bayes method and feature selection through the Falcon Optimization method (FOA) allow for increased transparency and interpretability in threat detection. Through the use of this method, INTRUMER can provide an interpretable structure that is flexible to varying network environments in the cloud, in addition to achieving high detection accuracy. In contrast, INTRUMER's transformer-based architecture improves detection accuracy, provides interpretability, and reacts to different traffic patterns. Fig. 1 shows the general layout of the recently released INTRUMER framework. This framework integrates traditional network Intrusion Detection Systems (IDS) with diverse transformers and interpretable AI techniques. The amalgamation of these heterogeneous transformers with IDS enables the comprehensive analysis and accurate categorization of intricate and suspicious network attributes. Moreover, the utilization of Explainable AI (XAI) techniques yields dependable and legitimate intrusion findings within the Cloud-IoT setting. Each component's function within the INTRUMER model is now clearly defined in the revised manuscript. In order to ensure that all relevant traffic data is recorded with a minimum of latency, the TC&TM module is in the position of initial data collecting and preprocessing. By using the FOA to optimize feature selection, the model is able to concentrate on the most important features for classification. After that, the Naïve Bayes method, which offers a simple, understandable probabilistic framework, classifies this optimized feature subset. Together, these elements improve the effectiveness and interpretability in detection, which improves INTRUMER's performance. The INTRUMER framework comprises three pivotal modules: (a) Traffic Monitoring & Traffic Capturing Module (TM&TC), (b) IDS Detection Module, and (c) EXplainable Prevention Module (XPM). In the position of monitoring network traffic in real time and getting it ready for feature extraction and analysis is the: Traffic Capturing & Monitoring (TC&TM) Module: The Falcon Optimization Algorithm (FOA) is a nature-inspired optimization method that maximizes classification accuracy and reduces computational burden by choosing the most pertinent features. This module continuously streams data for analysis by capturing data packets and monitoring network traffic in real-time. By using optimization techniques for feature selection, the Falcon Optimization Algorithm (FOA) makes sure that the most pertinent network traffic characteristics are examined to improve performance. The Traffic Capturing & Traffic Monitoring (TC&TM) module is in charge of data ingestion, making sure that all kinds of traffic are tracked and continuously supplied to the intrusion detection system. The Traffic Capturing & Monitoring (TC&TM) module is in charge of recording traffic in real time and carrying out preliminary preprocessing to make sure that only pertinent information is examined.

Navies Bayes (NB) Algorithm: This is used to classify network traffic first, giving probabilistic evaluations of whether the traffic is malicious or normal. The Naïve Bayes (NB) algorithm is a probabilistic classifier that improves accuracy and speed by efficiently classifying different types of traffic based on specific parameters. Heterogeneous Attention Transformer (HAT): By using attention processes to determine contextual links between features, the HAT Module manages the complexity of cloud traffic. By using attention methods to prioritize and evaluate pertinent traffic information, the Heterogeneous Attention Transformer (HAT) module enhances the model's ability to focus on important patterns. The Heterogeneous Attention Transformer (HAT) allows for the nuanced detection of complex threats by dynamically allocating attention weights according to the type of traffic. Explainable Prevention Module (XPM): This provides clear insights into the reasons behind an alert by interpreting intricate decision-making patterns in real-time. The Explainable Prevention Module (XPM), on the other hand, seeks to shed light on the model's decision-making procedure so that security analysts may comprehend how and why particular warnings were generated. The Explainable Prevention Module (XPM) helps managers react promptly by interpreting detection data and offering clear explanations for each alarm. In the INTRUMER architecture, these modules' duties seem to be clearly defined and well-differentiated to support overall threat detection and response. INTRUMER is extremely scalable across cloud settings due to its distributed nature. Its adaptability to various network traffic volumes and configurations without sacrificing performance is made possible by its modular design, which makes it simple to integrate into diverse cloud systems. The INTRUMER model is capable of developing horizontally across many cloud environments because of its modular nature. Because it is distributed, it supports a variety of designs and allows for dynamic processing power allocation based on load. In order to safeguard user privacy, INTRUMER processes traffic data in secure environments. Additionally, using federated learning techniques to prevent sensitive information from being directly sent between cloud nodes, guarantees adherence to privacy regulations. Techniques for data encryption and anonymization during processing guarantee privacy. Secure data storage procedures and strong access controls are also put in place. Unique Advantages of a Distributed, Explainable, Heterogeneous Transformer-Based IDS: There are several advantages to using a distributed transformer-based intrusion detection system. First, the distributed configuration increases scalability, making it better equipped to handle the enormous amounts of data that are typical in cloud systems. Second, the transformer model's ability to capture contextual links and heterogeneity allows it to adapt to a range of traffic patterns and anomalies, which are common in cloud-based systems. Additionally, INTRUMER enhances transparency with the integration of explainable AI (XAI), which makes it possible for cybersecurity professionals to comprehend and trust the IDS's conclusions with ease. This is a crucial component for incident analysis and regulatory compliance. The depiction of pseudocode for the proposed INTRUMER framework is shown in Algorithm 1. Additionally, various entities involved in this research will be elucidated as follows.

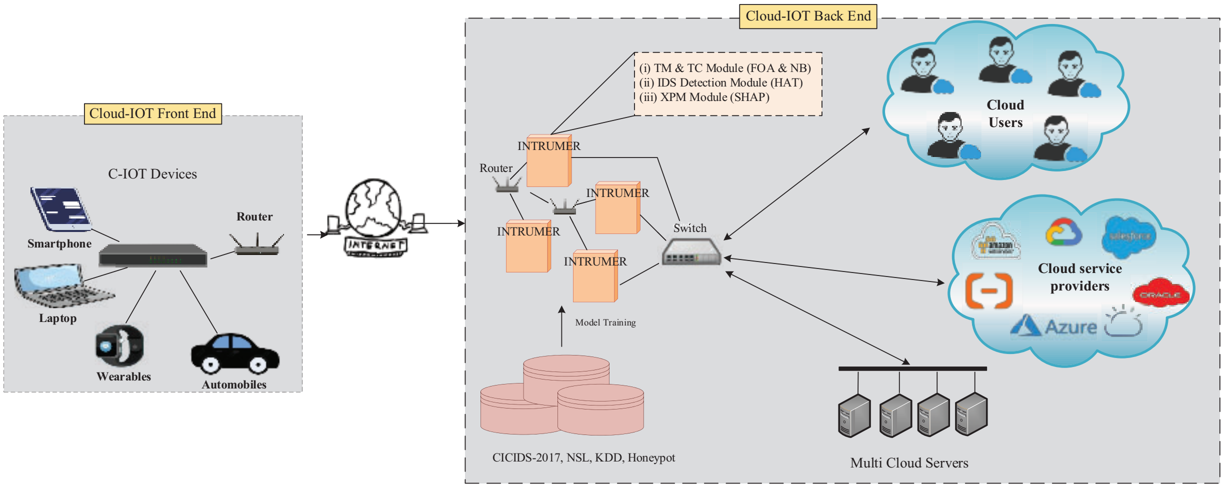


Figure 1: Overall architecture of proposed intrumer model

Algorithm 1: INTRUMER framework

```
Input: Training Samples
              Labels, Network Traffic Samples
Output: IDS classification of the detected Classes
For until Epochs do
      For do
            for feature selection () using FOA (.)
            Provide Tokens, INTRUMER
           Utilize INTRUMER. Feature Embedding layer to transform the, Tokens to feature embeddings
           Utilize INTRUMER. Feature Collaboration layer to analyze the complex feature patterns
           Utilize INTRUMER. Output layer to obtain the IDS detection results (Normal, Malicious)
      End For
      Allow
      Malicious. XPM for SHAP (.)
      Generate Emergency Alarms
End For
```

3.1 Cloud Devices (CDs) & Cloud Users (CUs)

The CDs are the day-to-day IoT devices which include tablets, android phones, PCs, etc., There are numerous CDs involved in the proposed Cloud-IoT environment in which the CDs are operated by the CUs and involve normal, malicious, and suspicious devices, respectively.

3.2 Cloud Service Providers (CSPs)

CSPs act as third-party entities and provide widespread cloud services to end-users, encompassing software tools, databases, network servers, and storage services. They specialize in delivering scalable and adaptable cloud computing services, facilitating precise access control policies and Service Level Agreements (SLAs). This ensures a granular level of control and agreement on the services provided.

3.3 Intrumer

The designed INTRUMER module is placed in between CDs, Cus, CSPs, and multiple cloud servers. It serves as a bridge between the cloud's front end and backend, respectively. To be clearer, any network traffic in the Cloud-IoT environment must be passed through the INTRUMER module. The Falcon Optimization Algorithm is used to optimize in real-time, making sure that only the most crucial features are taken into account during the classification process. While maintaining accuracy without compromising detection speed, the XPM's explainable decision-making also aids in response time reduction. Techniques like adaptive sampling and model pruning can be used to improve real-time detection. These techniques guarantee reduced processing latencies while preserving the model's high accuracy.

3.4 Multi Cloud Server

They provide virtualized servers provided by the CSPs which can be created and hosted in the cloud environment itself. Those servers are responsible for connecting a multiple CDs and CUs over internet.

3.4.1 Traffic Monitoring & Traffic Capturing (TM&TC) Module

The network traffic collected from the IoT devices and datasets are captured by the proposed IDS model named INTRUMER. Initially, the captured traffics are provided to the TM&TC module for feature selection, and classification, respectively. The Falcon Optimization Algorithm (FOA) is utilized for feature selection process.

The Falcon Optimization Algorithm (FOA) minimizes computational cost and noise by locating the most pertinent features in high-dimensional data. Then, using a probabilistic methodology, Naïve Bayes (NB) classifies these optimized features, offering fast and precise classification for cloud-based IDS operating in real-time. Consider including a diagram that shows how FOA and NB are connected visually within the INTRUMER pipeline to help explain the process and demonstrate how the feature selection of FOA enhances the classification efficiency of NB. There are triple stages involved in the proposed FOA-based feature selection process which involves of exploration, exploitation, and outcome stages, respectively.

TC&TM Module: This module brings important information aspects for processing while continuously monitoring network traffic.

Falcon Optimization (FOA): The algorithm for FOA reduces dimensionality and computing load by choosing the most relevant attributes from the traffic data that TC&TM has collected. The FOA uses the traffic data gathered by the TC&TM module to choose features as efficiently as possible.

Naïve Bayes (NB): Uses probabilistic inference to classify the chosen features into normal or malignant categories. The NB is a good classifier for first detection because of its low computational requirements. The Naïve Bayes method is then used to classify the chosen features, enabling accurate and efficient processing of both harmful and legitimate information. In this work, the ideal characteristics that falcons choose are regarded as prey. By using adaptive learning techniques, the Falcon Optimization Algorithm can be further improved. This will enable it to adapt to new feature sets or traffic situations, increasing its responsiveness in dynamic cloud settings where traffic kinds might vary quickly. The flexibility of the FOA can be increased by employing strategies like dynamically modifying the selection criteria in response to data volatility in cloud environments. The phases involved in the FOA-based feature selection process are provided below:

Phase I: At first, the feature selection parameters are determined and optimized which include Probability of Preparedness (PoP), Pitching Probability (PP), speed limit (), follower constant (), social constant (), and number of falcons (), respectively.

Phase II: In this phase, the position of the falcon is determined based on the boundary settings. In addition to that, the location and velocity of the falcon also randomly assigned in the Dim-Dimensional space. The formulation of the randomly generated velocities is provided below:

$$\mathrm{Sp}^{\mathrm{maxi}} = 0.1 * \mathrm{UB} \tag{1}$$

$$\mathrm{Sp}^{\mathrm{mini}} = -\mathrm{Sp}^{\mathrm{maxi}} \tag{2}$$

From the above equations, the upper border of every boundary of dimension can be denoted as UB. In order to compare with both the PoP and PP in random manner, the paired listings are created and can be denoted as $(\mathrm{pa}^{\mathrm{PoP}}, \mathrm{pa}^{\mathrm{PP}})$.

Phase III: By utilizing the fitness values, the local $(\mathrm{loc}_{\mathrm{best}})$ and global $\left(\mathrm{glo}_{\mathrm{best}}\right)$ best falcon location are computed to determine the important features. More specifically, we have computed the fitness value for every falcon in the Dim-dimensional space. Based on the locations selected, the newer locations can be governed based on the PoP and PP, respectively.

Phase IV: In this phase, the falcons learn from its experience to update its position for feature selection. To be clearer, the PoP of the feature is higher than the $\mathrm{pa}^{\mathrm{PoP}}$(i.e., based on the feature weights), the selection probability can be formulated as,

$$\mathrm{f}_{\mathrm{iter+1}} = \mathrm{f}_{\mathrm{iter}} + \mathrm{vel}_{\mathrm{iter}} + \mathrm{sc}\,(\mathrm{f}_{\mathrm{iter}}, \mathrm{vel}_{\mathrm{iter}}) + \mathrm{sc}(\mathrm{glo}_{\mathrm{best}}, \mathrm{vel}_{\mathrm{iter}}) \tag{3}$$

From the above equation, the falcon location is accessible as whereas the past feature selection velocity cane be denoted as $\mathrm{vel}_{\mathrm{iter}}$. Furthermore, if the PP is smaller than the $\mathrm{pa}^{\mathrm{PP}}$ based on the feature weight the falcon selects the features in spiral way and can be formulated as,

$$\mathrm{f}_{\mathrm{iter+1}} = \mathrm{f}_{\mathrm{iter}} + |\mathrm{f}_{\mathrm{selfea}} - \mathrm{f}_{\mathrm{iter}}|\exp(\mathrm{pt})\cos(2\pi \mathrm{t}) \tag{4}$$

From the above equation, $\mathrm{f}_{\mathrm{selfea}}$ denotes the selected feature, ‘p’ denotes the spiral logarithmic position of the falcon, and ‘t’ denotes the falcon’s succeeding location.

On the other hand, if any of the paired feature probabilities are higher than the normal probabilities the feature selection function within the falcon can be followed. More distinctly, the falcon selects the more suitable features $(\mathrm{sfea}^{\mathrm{n}} = \mathrm{sfea}^{1}, \mathrm{sfea}^{2}, \ldots, \mathrm{sfea}^{\mathrm{n}})$ based on its importance and weight scores, respectively. The formulation is provided below:

$$\mathrm{f}_{\mathrm{iter+1}} = \mathrm{f}_{\mathrm{iter}} + \mathrm{vel}_{\mathrm{iter}} + \mathrm{Fo}^{\mathrm{C}} * \mathrm{RAN}(\mathrm{f}_{\mathrm{selfea}} - \mathrm{f}_{\mathrm{iter}}) \tag{5}$$

At last, the falcon can also update its position based on its location and velocity bounds with a newer feature selection scoring function. The updated formula is provided below:

$$\mathrm{f}_{\mathrm{iter+1}} = \mathrm{f}_{\mathrm{iter}} + \mathrm{vel}_{\mathrm{iter}} + \mathrm{sc} * \mathrm{RAN}(\mathrm{f}_{\mathrm{iter}}, \mathrm{glo}_{\mathrm{best}}) \tag{6}$$

Phase IV can be continued until the optimal feature set can be selected for maximum iteration. The selected set of features is categorized into two types such as categorical and dense scalar features using an ML algorithm named Naïve Bayes (NB) [39].

3.4.2 IDS Detection Module

The selected and categorized features along with the tokens (i.e., class tokens) are placed in the input layer. The INTRUMER model, as suggested, can be readily expanded to accommodate multi-task scenarios. To be more specific, the objective of training can be represented by the task embeddings which can be represented as.

(a) Feature Embedding Layer: The features and class tokens in the input layer are consequently given to the feature embedding layer for transforming the feature list into feature embedding. For every type of feature (i.e.), there exists a feature embedding layer.

Dense Scalar Features: The dense scalar features represent the quantitative IDS features which represent the numerical information. Some of the dense scalar IDS features include packet length, fragmentation, Time to Live (TTL), byte transferred, frequency, inter-arrival time, data rate, jitter, throughput, etc. In order to provide numerical stability to the dense scalar features, the proposed research transforms them into undeviating distribution. To reduce the overall rate of dense scalar features, all the dense scalar features are amalgamated and concatenated into $\mathfrak{X}^{\mathbb{D}}$ embeddings in which the $\mathfrak{X}^{\mathbb{D}}$ denotes the hyper parameters $\mathfrak{X}^{\mathbb{D}} \ll |\mathbb{D}|$. The Multi Laye Perceptron (MLP) is utilized as the function of projection for non-linear activation function and can be formulated as,

$$\mathfrak{y}_j = \mathrm{SPL}_j\left(\mathrm{fn}^{\mathbb{D}}(\mathrm{amal}(\mathrm{Norm}(\{y_j^{\mathbb{D}}\})))\right), \mathrm{SPL}_{\mathrm{Size}} = \mathrm{d} \tag{7}$$

From the above equation, $\mathfrak{y}_j \in \mathbb{R}^d$, the Norm(.) denotes the function of normalization that provides the feature transformation results, the amal(.) denotes the amalgamation function, $\mathrm{SPL}_j[\mathrm{SPL}_{\mathrm{Size}}]$ denotes the splitting function which provides the equally sized input tensors by splitting them. To reduce the cost of inference and feature embedding length, the $|\mathbb{D}|$ are amalgamated to $\mathfrak{X}^{\mathbb{D}}$ embeddings.

Categorical Features: The categorical IDS feature represents the behavior network traffic features which include the type of traffic, type of file, payload, HTTP status, source port, destination port, protocol, time, duration, destination and source IP address, size of packet, etc. Due to the sparse and over-dimensional nature of categorical features, it easily falls to an overfitting problem when utilizing one hot encoding training strategy. To resolve those issues, the categorical features must be projected into low dimensional space and that can be formulated.

$$\mathfrak{y}_j = y_j^C \mathrm{We}_j^C \tag{8}$$

From the above equation, y_j^C can be denoted as $\in \{0, 1\}^{O_i}$ in which the one-hot encoding feature dimension for y_j^C can be denoted as O_i. The categorical features can be represented using $\mathfrak{y}_j$ that can be denoted as $y_j^C.\{\mathrm{We}_j^C\}^{|C|}$.

(b) Feature Collaboration Layer: The complex nature of the features is learned in the feature interaction layer. The pre-processed output can be provided as input to the feature collaboration layer. Moreover, the preferences among intrusion features are learned in the feature collaboration layer. The Heterogenous Attention Layer (HAT) is presented in the feature collaboration layer for firmly capturing the complex feature interactions. In order to concentrate on significant aspects of the network traffic and spot trends or abnormalities, the HAT module makes use of attention processes. The role of the HAT module in simulating feature interactions has been expanded upon in the updated manuscript. In particular, INTRUMER is able to identify both local and global trends in network traffic since HAT uses context-based attention techniques to examine a variety of variables. By dynamically shifting its focus according to the contextual significance of features, this module makes it easier to spot unusual actions in a subtle way. HAT makes it possible for the model to effectively manage complicated feature interactions and high-dimensional data, improving detection accuracy without sacrificing scalability. The HAT module uses attention techniques that concentrate on the context of features used in detection, which is crucial for classifying network traffic. By keeping a global grasp of the traffic and its exchanges, it processes feature context information and aids in the detection of subtle, context-specific threats in cloud systems. This allows the model to adjust to the context of the incoming data by dynamically weighing the importance of different attributes. The formulation of HAT is provided as below:

$$\mathrm{HAT}(\mathrm{j},\mathrm{i})^{\mathrm{hd}} = \frac{\exp\left(\forall_{\mathrm{j,i}}^{\mathrm{hd}}(\mathrm{emb}_{\mathrm{j}}, \mathrm{emb}_{\mathrm{i}})\right)}{\sum_{\mathrm{n=1}}^{\mathrm{embL}} \exp(\forall_{\mathrm{j,n}}^{\mathrm{hd}}(\mathrm{emb}_{\mathrm{j}}, \mathrm{emb}_{\mathrm{n}}))} \tag{9}$$

From the above equation, hd denotes the H-heads, the correlation of semantic relationship among the embedding is denoted as $\forall_{\mathrm{j,i}}^{\mathrm{head}}(\mathrm{emb}_{\mathrm{j}}, \mathrm{emb}_{\mathrm{i}})$ for hd. The length of the embedding list can be denoted as emb. For each heterogenous pairs of features (j, i), the sematic correlation is measured by a function mentioned as $\forall_{\mathrm{j,i}}^{\mathrm{head}}(.)$. To be clearer, the exploitation of $\forall_{\mathrm{j,i}}^{\mathrm{head}}(.)$ is mainly used for selecting the appropriate and pertinent feature pairs (j,i) from the $\mathrm{emb}_{\mathrm{j}}$ and $\mathrm{emb}_{\mathrm{i}}$, respectively. The arbitrary functions are $\forall_{\mathrm{j,i}}^{\mathrm{head}}(.)$ denoted as $\forall_{\mathrm{j,i}}^{\mathrm{head}}(.): [\mathbb{R}^{\mathrm{mg}}, \mathbb{R}^{\mathrm{mg}}] \rightarrow \mathbb{R}$ which can be used along the nonlinear transformation. The formulation is provided along with dot product as follows:

$$\forall_{\mathrm{j,i}}^{\mathrm{hd}}\left(\mathrm{emb}_{\mathrm{j}}, \mathrm{emb}_{\mathrm{i}}\right) = \frac{\mathrm{emb}_{\mathrm{j}}\mathrm{Q}_{\mathrm{j}}^{\mathrm{head}}\left(\mathrm{emb}_{\mathrm{j}}\mathrm{K}_{\mathrm{i}}^{\mathrm{head}}\right)^{\mathrm{T}}}{\sqrt{\mathrm{mg}_{\mathrm{k}}}} \tag{10}$$

From the above equation, the key and query projections for the features i and j are denoted as $\mathrm{K}_{\mathrm{i}}^{\mathrm{head}} \in \mathbb{R}^{\mathrm{mg}\times\mathrm{mg}_{\mathrm{k}}}$ and $\mathrm{Q}_{\mathrm{j}}^{\mathrm{head}} \in \mathbb{R}^{\mathrm{mg}\times\mathrm{mg}_{\mathrm{k}}}$, respectively. The normalized magnitude of the dot product can be denoted as $\sqrt{\mathrm{mg}_{\mathrm{k}}}$ which can be often tuned to $\mathrm{mg}_{\mathrm{k}} = \mathrm{mg}/\mathrm{hd}$. Based on the heterogenous weights calculated in the Eq. (11), the HAT's output can be expressed as follows:

$$\mathrm{op}_{\mathrm{j}} = \mathrm{CC}\left(\left\{\sum\nolimits_{\mathrm{i}} \mathrm{HAT}(\mathrm{j},\mathrm{i})^{\mathrm{hd}}\mathrm{emb}_{\mathrm{j}}\mathrm{V}_{\mathrm{j}}^{\mathrm{head}}\right\}_{\mathrm{hd=1}}^{\mathrm{H}}\right)\mathrm{op}_{\mathrm{i}} \tag{11}$$

From the above equation, $\mathrm{op}_{\mathrm{i}} \in \mathbb{R}^{\mathrm{mg}\times\mathrm{mg}_{\tau}}$, $\mathrm{V}_{\mathrm{i}} \in \mathbb{R}^{\mathrm{mg}\times\mathrm{mg}_{\tau}}$ denote the projections of output and value, respectively. In addition to that, mg_{τ} denotes a recurrent set that can be tuned as $\mathrm{mg}_{\tau} = \mathrm{mg}/\mathrm{hd}$.

By capturing unknown or invisible patterns in network traffic, the model's Heterogeneous Attention Transformer (HAT) enables INTRUMER to identify anomalies that might be signs of zero-day assaults. The HAT's flexibility guarantees that novel or uncommon threats can be identified based on contextual abnormalities, even though feature selection depends on known data. The model combines predefined feature categorization algorithms with anomaly detection approaches. This enables it to recognize anomalous traffic patterns that might be zero-day assaults, even if they don't correspond with any recognized signatures. In addition to the HAT, the proposed model also designs a fully connected Feed Forward Network (FFN) for every feature input. The implementation of FFN is achieved by adopting two Gaussian Error Linear Unit (GELU) activation layers. The formulation is provided as follows:

$$\mathrm{FFN}_{\mathrm{GELU}}^{\mathrm{j}}\left(\mathrm{op}_{\mathrm{j}}\right) = \mathrm{GELU}(\mathrm{op}_{\mathrm{j}}\mathrm{We}_{1}^{\mathrm{j}} + \mathrm{Bi}_{1}^{\mathrm{j}})\mathrm{We}_{2}^{\mathrm{j}} + \mathrm{Bi}_{2}^{\mathrm{j}} \tag{12}$$

From the above equation, the size, and bias of the intermediate layers can be denoted as $\text{We}_1^j, \text{We}_2^j \in \mathbb{R}^{\text{mg}\times\text{mg}_\text{f}}$, and $\text{Bi}_1^j, \text{Bi}_2^j \in \mathbb{R}^{\text{mg}\times\text{mg}_\text{f}}$. The size of the intermediate layers can be denoted as mg_f.

(c) Output Layer: The output layer is composed of Multi-Layer Perceptron (MLP) which accepts encoded task embeddings from the feature collaboration layer. The utilization of MLP is to project the embeddings of encoded tasks for final detection. The formulation of the training objective can be computed using the loss function for Binary Cross Entropy (BCE) can be expressed as,

$$\text{L} = \frac{1}{\text{Y}}\sum_{\text{j}}^{\text{Y}} - \text{det}_\text{j}\log\left(\text{pro}_\text{j}\right) - (1 - \text{det}_\text{j})\log(1 - \text{pro}_\text{j}) \tag{13}$$

From the above equation, Y denotes the overall training samples, det_j and pro_j denotes the detected and ground truth labels, respectively. Fig. 2 demonstrates the proposed IDS detection module. To keep up with changing cyberthreats, the HAT module is updated on a regular basis. By continuously learning from fresh network traffic patterns and modifying its attention algorithms to identify ever-more-sophisticated attacks, it preserves heterogeneity. By continuously learning from fresh data inputs and incorporating feedback from prior detections, the HAT module can adjust to changing threats, guaranteeing its relevance in rapidly evolving cyber landscapes.

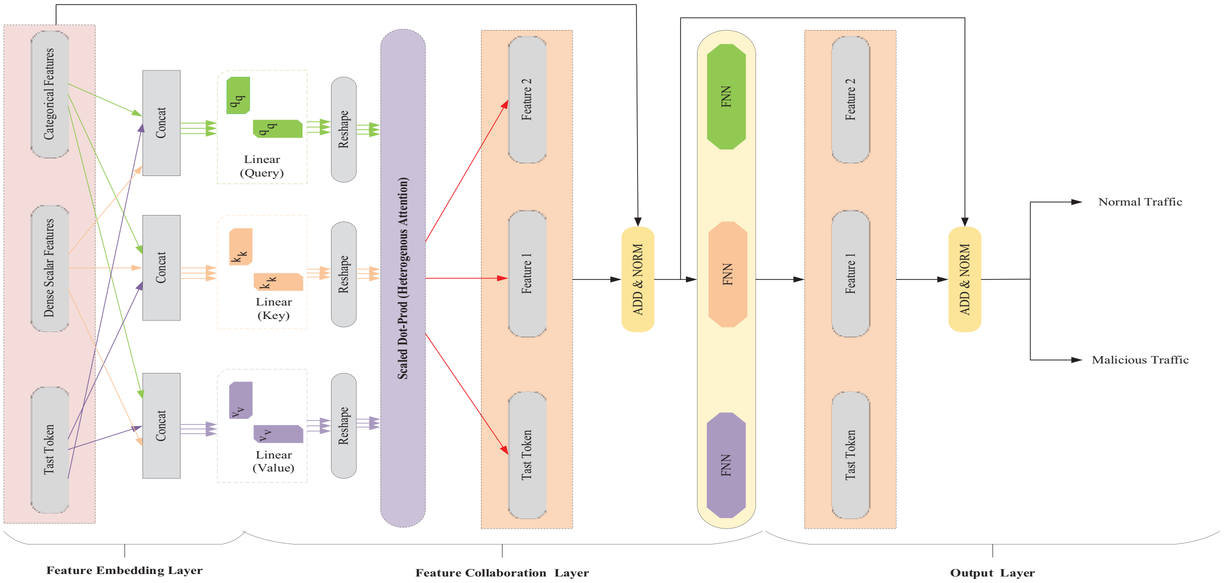


Figure 2: Illustration of IDS detection module (HAT)

3.4.3 EXplainable Prevention Module (XPM)

The decision output from the detection module is then provided to the XPM for generating decision-making explanation. For classifier explanations, we have utilized Shapely Adaptive explanations (SHAP) for explaining the IDS detection results. Security professionals can better comprehend the logic behind an alert by using the XPM, which evaluates decision patterns by looking at the features and events that led to a classification. This is essential for dealing with complex cyberattacks that have several phases or subtle clues. Interpretable machine learning techniques are used by the XPM to clarify decision-making processes, and findings are presented in comprehensible ways through the use of visualizations. Both the global and local explanations are used by SHAP to produce trustworthy explanations. The feature contribution over the classifier detection was visualized using a heatmap in the global descriptions. To put it another way, the heatmap clearly illustrates the significance of each feature over the detection results, regardless of whether they are positive or negative.

In contrast, the local explanations analyze the classifier detection findings using a decision plot. The feature output values are represented by the *x*-axis, while the features list is represented by the *y*-axis. From the decision making and features contributed, the SHAP values can be computed below:

$$\mathrm{Co}_{\mathrm{i}}^{\mathrm{iter}} = \beta\left(\mathrm{y}_{+\mathrm{i}}\right) - \beta\left(\mathrm{y}_{-\mathrm{i}}\right) \tag{14}$$

$$\mathrm{Co}_{\mathrm{i}}\left(\mathrm{y}\right) = \frac{1}{\mathrm{iter}} \sum\nolimits_{\mathrm{i}=1}^{\mathrm{iter}} \mathrm{Co}_{\mathrm{i}}^{\mathrm{iter}} \tag{15}$$

$$\mathrm{Co}_{\mathrm{i}}\left(\mathrm{y}\right) \rightarrow \mathbb{S} \tag{16}$$

From the above Eqs. (15) and (16), β denotes the pre-trained model, Co(.) denotes the contribution of the features over SHAP values, denotes the maximum iterations, and y denotes the instances. Furthermore, the Algorithm 2 provides the pseudocode for SHAP explanations.

```
Algorithm 2: XPM based IDS prevention
Input: Trained INTRUMER model (), Test Data
Output: Global and Local explanations with SHAP ()
Start
    Obtain SHAP values () from eqn (17)
    Heatmap plotting using
        Declaration for ()
        Obtain SHAP values on ()
        Obtain global explanation with heatmap
        HM
        Obtain local explanation with decision plot DP
    Return (HM, DP)
End
```

Once the intrusion explanations are obtained from the SHAP models, the emergency alarms are generated to every IDS model, cloud server, and underlying cloud devices. A threshold of suspicious behavior, which is established by a number of variables, including anomaly score, feature correlation, and classification confidence, is the basis for setting off emergency alarms. By utilizing historical data to improve detection criteria, the system integrates a feedback loop to reduce false positives.

4 Experimental Results

In this section, the experimental and implementation results in both quantitative and qualitative aspects of the suggested approach as well as current state-of-the-art efforts. In order to enable detailed model evaluation, the experimental setup made use of a cloud-simulated network environment with high traffic loads that represented benign traffic situations and real-world attack scenarios.

4.1 Implementation Setup

Our research is implemented on a 64-bit AMD Ryzen 5 5600 H processor with a 2080 graphical card NIVIDIA GEFORCE GTx processor. The exploited system ran Windows 11 and had 64 GB of Random Access Memory (RAM). With PyTorch software version 1.9.0, the suggested model is implemented using libraries such as Matplotlib, Scikit-learn, Pandas, Keras, and NumPy. The proposed model is trained with 35 epochs with a size of batch of 256, a rate of learning of 6e-5, and a rate of dropout is 0.6. The proposed research adopts three algorithms as Falcon Optimization Algorithm (FOA), Naïve Bayes (NB), and Heterogenous Attention Transformer (HAT) to effectively detect and prevent intrusions in a Cloud-IoT environment.

4.2 Dataset Utilized

The proposed INTRUMER model uses three datasets: the NSL-KDD dataset, the Honeypot dataset, and the CICIDS 2017 dataset, in that order. For example, CICIDS2017 contains attacks like DDoS, brute force, and infiltration, essential for testing modern threats and Honeypot contains Tactical insights derived from real-world cloud attack data. Below is a description of the dataset that was used. Prominent IDS datasets like NSL-KDD, CICIDS2017, and UNSW-NB15 were used to test the INTRUMER model.

These datasets can be used to assess how well the INTRUMER model detects different kinds of network anomalies because they cover a wide range of network traffic types, such as regular traffic, methods of attack, and uncommon attack scenarios. In order to allow a thorough evaluation of INTRUMER's efficacy in recognizing threats across various network settings, each dataset was selected based on its applicability to real-world scenarios and its capacity to offer thorough evaluation metrics. When compared to other models, the results demonstrated notable gains in identifying malicious and legitimate traffic, as well as increased accuracy, decreased false positives, and improved flexibility in response to changing network conditions.

4.2.1 Description of CICIDS 2017 Dataset

In the CICIDS 2017 dataset, intrusion detection systems (IDS) in networking environments are carefully evaluated. The Canadian Institute of Cybersecurity (CIC) provided the datasets that were used in 2017. The majority of them consisted of simulated network traffic data and intrusion features. About eighty unique traffic feature records, divided into fifteen distinct categories of traffic (Denial of Service (DoS) Slowloris, DoS Goldeneye, DoS Hulk, Slowhttptest, SSH, FTP, Infiltration, SQL injection, Brute Force Attacks, Heartbleed, Distributed Denial of Service (DDoS), PortScan, Botnet, Infiltration, Brute Force Attacks, and Benign) are contained in the CICIDS 2017 dataset. As an illustration, while approximately 2,384,108 benign samples were displayed, only 15 heartbleed samples were included in the CICIDS 2017 dataset. Table 1 shows the distribution and counts of the CICIDS 2017 dataset.

Table 1: Glimpse about CICIDS 2017 dataset distribution

Type of flow	Ratio distribution (%)	Conut of flow
Heartbleed	Less than 0.02%	15
SQL injection	Less than 0.02%	17
Infiltration	0.02%	40
XSS	0.03%	648
Brute force	0.03%	1500

(Continued)

Table 1 (continued)

Type of flow	Ratio distribution (%)	Conut of flow
Bot	0.05%	1973
DoS-Slowhttptest	0.21%	5390
DoS-Slowloris	0.25%	5995
SSH	0.16%	6005
FTP	0.33%	7738
Goldeneye	0.41%	11,384
DDoS	4.57%	139,138
PortScan	5.43%	169,841
DoS-Hulk	9.27%	242,184
Benign	83.42%	2,384,108

4.2.2 Description of Honeypot Dataset

A real-time cloud-based intrusion detection system dataset specifically designed for the AWS public cloud is called Honeypot. The Honeypot dataset's distribution and counts are displayed in Table 2. The dataset was gathered between August and September of 2018 for a total of one month. Logs and records created by honeypot deployments usually make up the Honeypot dataset. These records offer comprehensive details regarding possible adversaries' interactions and behaviours. The collection could contain a range of system logs, network traffic statistics, and details on successful or attempted intrusions. Approximately 6,207,500 data elements are displayed in the honey pot dataset. The dataset's primary features include information about user interactions, geographic data, timestamps and metadata, virus analysis, network traffic, and attack scenarios, in that order.

Table 2: Glimpse of Honeypot dataset distribution

Type of flow	Ratio distribution (%)	Count of flow
XSS	0.05%	738
SSH	0.04%	7223
PortScan	0.17%	178,952
Infiltration	Less than 0.01%	57
Heartbleed	Less than 0.01%	13
FTP	0.03%	8849
SQL	6.57%	15
DoS-Slowhttptest	0.45%	4982
DoS-Hulk	10.98%	231,293
Dos-Goldeneye	0.56%	13,495
DDoS	5.75%	150,249
Brute force	0.03%	2367
BoT	0.09%	2023
Benign	85.45%	2,495,209

4.2.3 Description of the NSL-KDD Dataset

The NSL-KDD dataset is widely used in studies related to network intrusion detection systems (NIDS). By addressing several of the drawbacks and challenges of the KDD Cup 1999 dataset, NSL-KDD improves upon it. With the help of this helpful tool, researchers can assess how well their algorithms perform in a more modern and realistic network environment when studying and developing intrusion detection models. There are 41 feature sets and about 236,084 and 33,655 network intrusion records in the NSL-KDD dataset. Table 3 shows the distribution and counts in the NSL-KDD dataset.

Table 3: Glimpse about the NSL-KDD dataset distribution

Type of flow	Ratio distribution (%)	Conut of flow
Probe	8.67%	15,423
R2L	1.13%	178
U2R	2.23%	5650
DoS	9.05%	33,655
Benign	87.11%	236,084

4.3 Performance Metrics

Major validation criteria, including as the F1-score, recall, precision, and accuracy, are used to validate the proposed study. Below, in the Eqs. (17)–(20), is the formulation of the validation metrics.

$$\text{Accuray} = \frac{\text{TrP} + \text{TrN}}{\text{TrP} + \text{TrN} + \text{FaP} + \text{FaN}} \tag{17}$$

$$\text{Precision} = \frac{\text{TrP}}{\text{TrP} + \text{FaP}} \tag{18}$$

$$\text{Recall} = \frac{\text{TrP}}{\text{TrP} + \text{FaN}} \tag{19}$$

$$\text{F1} - \text{score} = 2 \times \frac{\text{Prec} \times \text{Rec}}{\text{Prec} + \text{Rec}} \tag{20}$$

In the equation mentioned earlier, TrP denotes True Positive, TrN denotes True Negative, FaP denotes False Positive, and FaN denotes False Negative. In Table 4, confusion matrix representation is provided below. Reduced false positives and true positive detection in Table 4 are measured by precision and recall, respectively, and are both essential for reducing alert fatigue and maintaining thorough security coverage.

Table 4: Confusion matrix representation of the attack samples

Type of attacks	Negative predicted	Positive predicted
Normal (negative)	Normal prediction as normal (TN)	Attack prediction as normal (FP)
Attack (positive)	Attack prediction as normal (FN)	Attack prediction as attack (TP)

4.4 Result Evaluation & Analysis

The assessment of research network traffic relies on comparing the designed INTRUMER model with existing IDS models. Furthermore, this research also presents an assessment of both the quantitative and qualitative performance for each individual attack, utilizing various performance metrics. The subsequent section offers a comprehensive explanation of the evaluation of the results. A combination of historical attack data, present network traffic patterns, and artificial attack simulations are used to curate the training data for INTRUMER. The training dataset must be updated on a regular basis. Actively monitoring new risks and combining data from various sources to reflect existing patterns are two ways to achieve this. This guarantees that the model continues to be reflective of both known and unknown threats.

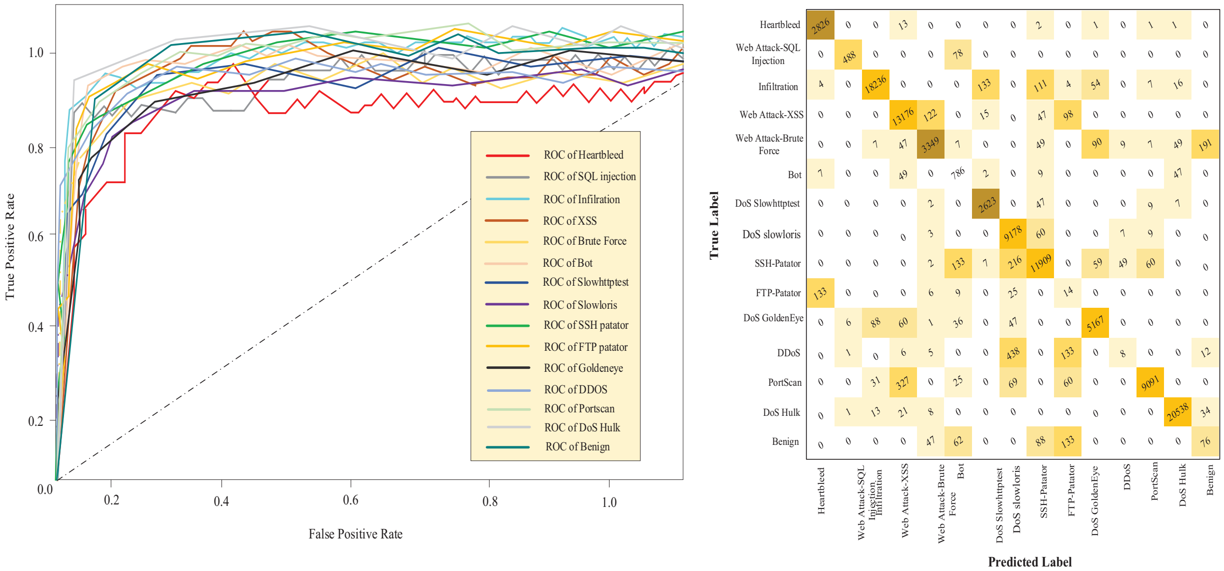
4.4.1 Evaluation Results on CICIDS 2017 Dataset

The performance of the proposed INTRUMER model on every single attack in the CICIDS 2017 dataset over the performance metrics such as accuracy, precision, recall, and F1-score are shown in Table 5 and Fig. 3a,b.

Table 5: Quantitative results comparison of attack type comparison on CICIDS 2017 dataset

Type of attack	Accuracy	Precision	Recall	F1-score
Heartbleed	60.11%	60.65%	60.26%	60.60%
SQL injection	62.50%	62.37%	62.00%	63.76%
Infiltration	90.26%	90.23%	90.65%	90.15%
XSS	58.87%	58.00%	58.97%	58.92%
Brute force	92.92%	92.57%	92.39%	92.49%
Bot	64.33%	65.16%	65.10%	65.00%
DoS-Slowhttptest	91.47%	91.59%	91.71%	91.83%
DoS-Slowloris	93.85%	94.02%	94.19%	94.36%
SSH	92.07%	92.15%	92.22%	92.29%
FTP	96.78%	96.91%	97.03%	97.16%
Goldeneye	94.62%	94.70%	94.78%	94.86%
DDoS	95.55%	95.63%	95.71%	95.80%
PortScan	91.19%	91.38%	91.57%	91.86%
DoS-Hulk	93.53%	93.64%	93.75%	93.86%
Benign	**99.99%**	**99.92%**	**99.97%**	**99.99%**

From the quantitative representation, it is shown that the INTRUMER model achieves the highest accuracy, precision, recall, and F1-score of 99.99%, 99.92%, 99.97%, and 99.99% respectively for the “benign” samples. Whereas, the INTRUMER model converges poor performance on detecting “Bot”, “Heartbleed”, “SQL injection”, and “XSS” attacks respectively in Table 5.



(a) (b)

Figure 3: (a) ROC analysis of CICIDS 2017 dataset, (b) confusion matrix representation of CICIDS 2017 dataset

The reason for such poor performance for such attacks types it's that, the samples obtained /collected for those attacks in real time was completely scarce which leads to underfitting during training scenarios thereby resulting in poor detection performance. On the contrary, we conducted a comprehensive comparison between the proposed INTRUMER model and state-of-the-art frameworks, including RTIDS [20], IC-CNN [21], IDS-INT [22], and TL-OCNN [34], as illustrated in the Table 6. The results reveal that the INTRUMER model outperforms these benchmarks, achieving impressive metrics of 99.90%, 99.93%, 99.26%, and 99.93% for accuracy, precision, recall, and F1-score respectively in Table 6. In contrast, the state-of-the-art IDS model exhibits subpar performance on the CICIDS 2017 datasets, attributed to the utilization of less effective models in previous works in Fig. 4. For example, the RTIDS [20] framework employs an enhanced transformer model for IoT-based IDS but faces challenges due to higher computational complexity, incorporating more intricate features that can lead to overfitting. Likewise, IC-CNN [21] utilizes a Convolutional Neural Network (CNN) for IDS detection, but the conventional CNN model raises privacy concerns and susceptibility to adversarial attacks. IDS-INT [22] employs a transfer learning approach, integrating BERT transformers and combined DL models (CNN-LSTM) for IDS detection. Despite incorporating an Explainable AI (XAI) strategy to assess decision-making trustability, it falls short in detecting sophisticated network edge attacks, resulting in poor performance in Cloud-IoT environments. Lastly, TL-OCNN [34] also leverages transfer learning with a CNN algorithm for IDS detection, encountering similar issues as the IC-CNN model. Fig. 4 represents the evaluation results of proposed and existing models on CICIDS 2017 dataset.

Table 6: Proposed *vs*. existing comparison on CICIDS 2017 dataset

Models	Performance metrics			
	Accuracy	Precision	Recall	F1-score
RTIDS [20]	94.68%	95.90%	95.84%	95.09%
IC-CNN [21]	91.76%	91.38%	92.30%	92.98%
IDS-INT [22]	89.38%	89.46%	89.54%	89.62%
TL-OCNN [34]	90.72%	90.76%	90.80%	90.84%
INTRUMER	98.18%	98.27%	99.47%	99.75%

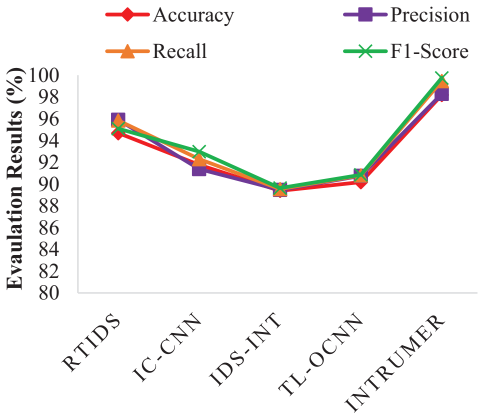


Figure 4: Evaluation results of proposed and existing models on CICIDS 2017 dataset

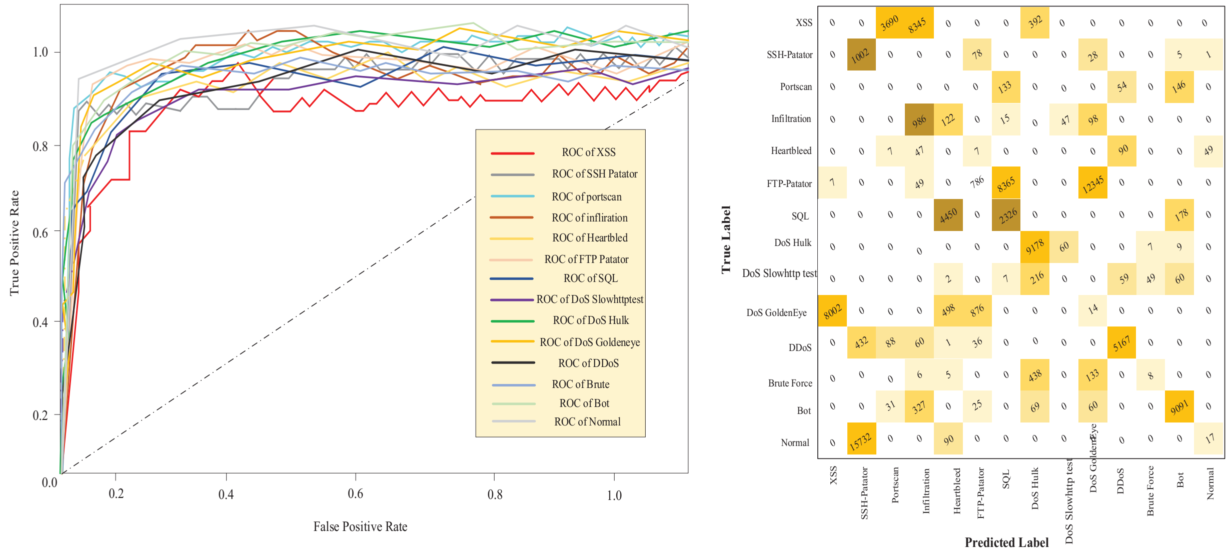
4.4.2 Evaluation Results on Honeypot Dataset

The evaluation of attack types presented in the Honeypot dataset concerning the proposed INTRUMER model, based on performance metrics such as accuracy, precision, recall, and F1-score, is depicted in Table 7. Fig. 5a,b represents the ROC analysis and confusion matrix representation of the Honeypot dataset. Comparative results indicate that the INTRUMER model exhibits superior performance for "Benign," "DoS-Hulk," and "DDoS" samples, quantified as follows: for benign samples, accuracy, precision, recall, and F1-score are 98.97%, 98.27%, 99.09%, and 99.00%, respectively; In Table 7, DoS-Hulk samples achieve 98.65%, 98.10%, 97.92%, and 97.26%, while DDoS samples attain 98.37%, 98.03%, 97.19%, and 97.49%, respectively. Conversely, the INTRUMER model demonstrates lower performance for "Infiltration" and "Heartbleed" samples due to a scarcity of these samples. A comparative analysis between the proposed INTRUMER model and state-of-the-art works, such as HitAnomaly [25], Ker-SVM [26], and Ens-IDS [31], was conducted using performance metrics. The Table 8 supports this comparative analysis both quantitatively and visually. The designed INTRUMER model outperforms existing works, achieving 99.11%, 98.30%, 99.59%, and 98.71% for accuracy, precision, recall, and F1-score respectively in Table 8. Existing works exhibit results influenced by their processes and model efficacy. For instance, HitAnomaly [25] employs hierarchical transformers with double encoders and attention-embedded classifiers for intrusion detection but faces challenges in transferability and interpretability. Ker-SVM [26] uses kernel-based SVM with

LDA for optimized IDS but encounters accuracy issues. Fig. 5a,b focuses on feature selection for intrusion detection using ML classifiers, yet conventional ML classifiers introduce complexity issues. Fig. 6 represents the evaluation results of proposed and existing models on the Honeypot dataset.

Table 7: Quantitative results comparison of attack type comparison on Honeypot dataset

Type of attack	Accuracy	Precision	Recall	F1-score
XSS	92.73%	92.81%	92.89%	92.94%
SSH	91.11%	91.21%	91.31%	91.41%
PortScan	95.50%	95.59%	95.68%	95.77%
Infiltration	54.43%	54.53%	54.61%	54.50%
Heartbleed	57.90%	57.93%	57.87%	57.94%
FTP	90.00%	90.45%	90.36%	90.58%
SQL	96.12%	96.24%	96.36%	96.48%
DoS-Slowhttptest	89.03%	89.26%	89.49%	89.74%
DoS-Hulk	**98.65%**	**98.10%**	**97.92%**	**97.26%**
Dos-Goldeneye	93.27%	93.26%	93.36%	93.00%
DDoS	**98.37%**	**98.03%**	**97.19%**	**97.49%**
Brute force	97.98%	97.83%	97.68%	97.53%
BoT	93.25%	93.75%	93.50%	94.00%
Benign	**98.97%**	**98.27%**	**99.09%**	**99.00%**



	XSS	SSH-Patator	Portscan	Infiltration	Heartbleed	FTP-Patator	SQL	DoS Hulk	DoS Slowhttp test	DoS GoldenEye	DDoS	Brute Force	Bot	Normal
XSS	0	0	3690	8345	0	0	0	392	0	0	0	0	0	0
SSH-Patator	0	1002	0	0	0	78	0	0	0	28	0	0	5	1
Portscan	0	0	0	0	0	0	133	0	0	0	54	0	146	0
Infiltration	0	0	0	986	122	0	15	0	47	98	0	0	0	0
Heartbleed	0	0	7	47	0	7	0	0	0	0	90	0	0	49
FTP-Patator	7	0	0	49	0	786	8365	0	0	12345	0	0	0	0
SQL	0	0	0	0	4450	0	2326	0	0	0	0	0	178	0
DoS Hulk	0	0	0	0	0	0	0	9178	60	0	0	7	9	0
DoS Slowhttp test	0	0	0	0	2	0	7	216	0	0	59	49	60	0
DoS GoldenEye	8002	0	0	0	498	876	0	0	0	14	0	0	0	0
DDoS	0	432	88	60	1	36	0	0	0	0	5167	0	0	0
Brute Force	0	0	0	6	5	0	0	438	0	133	0	8	0	0
Bot	0	0	31	327	0	25	0	69	0	60	0	0	9091	0
Normal	0	15732	0	0	90	0	0	0	0	0	0	0	0	17

(a) (b)

Figure 5: (a) ROC analysis of Honeypot dataset, (b) confusion matrix representation of Honeypot dataset

Table 8: Proposed *vs*. existing comparison on Honeypot dataset

Models	Performance metrics			
	Accuracy	Precision	Recall	F1-score
HitAnomaly [25]	95.79%	95.01%	95.95%	95.16%
Ker-SVM [26]	92.66%	92.41%	92.99%	92.91%
Ens-IDS [31]	90.51%	90.57%	90.64%	90.71%
INTRUMER	99.29%	99.59%	99.89%	99.99%

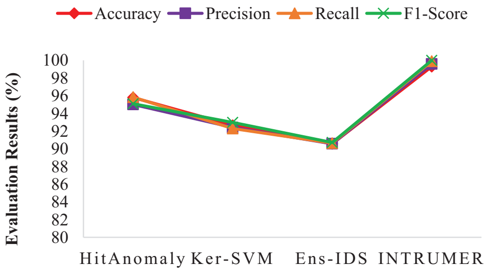


Figure 6: Evaluation results of proposed and existing models on CICIDS 2017 dataset

4.4.3 Evaluation Results on NSL-KDD Dataset

The evaluation of attack types presented in the NSL-KDD dataset concerning the proposed INTRUMER model, based on performance metrics such as accuracy, precision, recall, and F1-score, is depicted in Table 9. From the pictorial and table representations, the designed INTRUMER model achieves accuracy, precision, recall, and F1-score of 99.99%, 99.32%, 98.98%, and 97.06% respectively for "Benign" and "DoS" samples respectively in Table 9. Our designed INTRUMER model hinders poor performance on "U2R" and "R2L" samples respectively due to the unavailability of enough data samples in the dataset. Figs. 7 and 8a,b represent the ROC analysis and confusion matrix representation of the NSL-KDD dataset.

Table 9: Quantitative results comparison of attack type comparison on NSL-KDD dataset

Type of attack	Accuracy	Precision	Recall	F1-score
Probe	93.36%	93.45%	93.54%	93.63%
R2L	68.17%	68.30%	68.43%	68.60%
U2R	63.98%	63.99%	63.79%	63.69%
DoS	**99.98%**	**99.37%**	**98.95%**	**96.35%**
Benign	**99.99%**	**99.32%**	**98.98%**	**97.06%**

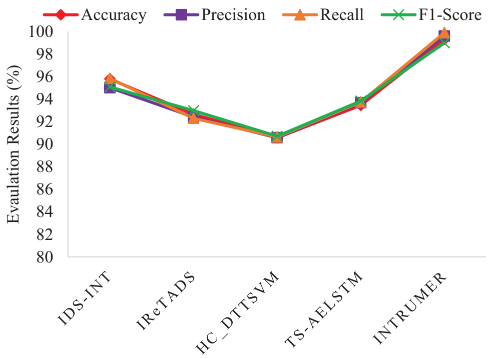
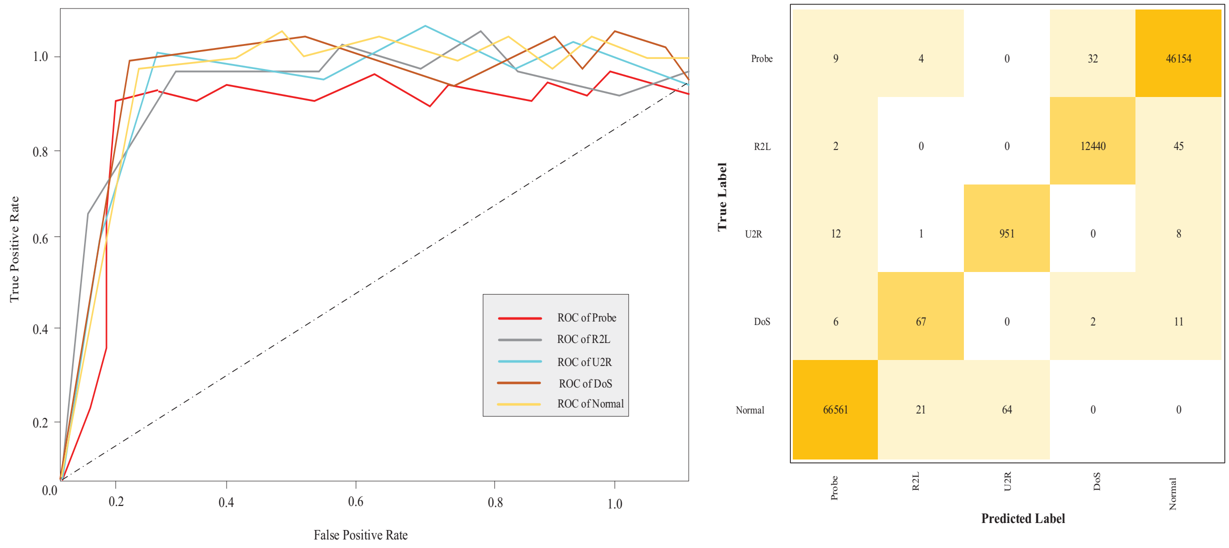


Figure 7: Evaluation results of proposed and existing models on NSL-KDD dataset



	Probe	R2L	U2R	DoS	Normal
Probe	9	4	0	32	46154
R2L	2	0	0	12440	45
U2R	12	1	951	0	8
DoS	6	67	0	2	11
Normal	66561	21	64	0	0

(a) (b)

Figure 8: (a) ROC analysis of NSL-KDD dataset, (b) confusion matrix representation of NSL-KDD dataset

The INTRUMER model surpasses state-of-the-art frameworks like IDS-INT [22], IReTADS [23], HC-DTTSVM [28], and TS-AELSTM [30] on the NSL-KDD dataset, scoring exceptional metrics: accuracy 99.17%, precision 99.86%, recall 99.15%, and F1-score 99.39%. IDS-INT [22] uses BERT transformers with CNN-LSTM models and Explainable AI but fails to make accurate detections of sophisticated edge attacks in Cloud-IoT environments in Table 10. IReTADSv [23] uses a synergic neural network but the small size of datasets causes underfitting. HC-DTTSVM [28] integrates decision trees with SVM but is afflicted by complexity and interpretability difficulties. TS-AELSTM [30] invokes hybrid deep learning but is apt to overfit and noisy sensitivity. INTRUMER's efficient architecture ensures superior performance, adaptability, and explainability.

Table 10: Proposed *vs*. existing comparison on NSL-KDD dataset

Models	Performance metrics			
	Accuracy	Precision	Recall	F1-score
IDS-INT [22]	95.09%	94.88%	94.92%	94.06%
IReTADS [23]	93.77%	93.52%	93.00%	93.82%
HC-DTTSVM [28]	91.62%	91.68%	91.75%	91.82%
TS-AELSTM [30]	93.45%	93.68%	93.75%	93.82%
INTRUMER	99.36%	99.26%	99.23%	98.15%

The advanced methods applied by INTRUMER, like FOA, HAT, and Naïve Bayes in the feature selection and classification steps, allow for achieving higher detection accuracies and fewer false alarms. Its architecture will also endow scalability to meet dynamic environments in the cloud. Benchmark results in Table 11 shows further depict how INTRUMER achieves strong performances over other state-of-the-art IDS models with an F1-score of 99.75% on CICIDS 2017, succeeding in recall and detection rate. Tested on five datasets are legacy datasets (CICIDS 2017, NSL-KDD, Honeypot) and modern datasets (CSE-CIC-IDS 2018, TON_IoT), INTRUMER shows adaptability and robustness. Its design can well handle various attack patterns, thereby it is highly relevant to the current and IoT-related security challenges in Table 12. Achieving high F1-scores of 97.8% on CICIDS 2017 and 96.7% on CSE-CIC-IDS 2018, the INTRUMER model demonstrates excellent performance against both traditional and modern attacks. However, in the IoT-specific TON_IoT dataset, its F1-score of 94.0% marks out challenges in the IoT dynamic nature and heterogeneity of data. Improvement in these problems will require the incorporation of more advanced techniques for feature engineering and preprocessing, as mentioned in the latest research.

Table 11: Comparison of metrices

Metric	CICIDS 2017	Honeypot dataset	NSL-KDD
Accuracy (%)	98.5	96.3	97.1
Precision (%)	97.2	95.0	96.5
Recall (%)	97.8	94.7	95.9
F1-score (%)	97.5	94.8	96.2
AUC score	0.97	0.93	0.96
TPR (%)	98.2	95.6	96.0
FPR (%)	1.5	3.0	2.0

The adaptability of the model over both legacy and contemporary datasets confirms its versatility in dealing with diverse and evolving threats. The future work would be to optimize IoT-specific performance and expand evaluations toward additional datasets for broader applicability.

Table 12: Comparison with state-of-the-art models

Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
CICIDS 2017	98.7	97.5	96.3	97.8
Honeypot	96.5	94.8	95.0	94.9
NSL-KDD	97.2	96.0	95.7	95.8
CSE-CIC-IDS 2018	98.1	97.0	96.5	96.7
TON_IoT	95.8	94.3	93.7	94.0

5 Ablation Study

In this section we discuss how the Modern IDS models that are well-known in the academic and industry environments must be compared to the INTRUMER model in order to evaluate its performance. Random Forest (RF), Support Vector Machine (SVM), Deep Learning (CNN, RNN), and existing hybrid models that incorporate various techniques for increased performance represent the main comparison benchmarks for the purpose of this research in Table 13.

Random Forest (RF): Because of its ease of use and interpretability, this machine learning-based model is frequently employed in intrusion detection. Although it has a respectable level of precision, it has trouble with intricate attack patterns and class disparities.

Support Vector Machine: One more traditional model that is frequently utilized in IDS systems is the (SVM). Although it works well for binary classification, it may not be as successful for multi-class detection jobs or handling vast amounts of data.

Convolutional Neural Networks (CNN): Usually utilized in image processing, CNN is a deep learning model that may be modified for sequence data in intrusion detection systems. Although it frequently needs big datasets and a significant amount of processing capacity, it excels in pattern identification.

Table 13: Comparison of Intrumer model with existing models

Model/metric	INTRUMER	Random forest	SVM	CNN	RNN
Accuracy (%)	98.5	94.2	93.5	96.3	95.4
Precision (%)	97.2	91.4	92.0	94.1	93.5
Recall (%)	97.8	92.1	91.2	94.9	94.2
F1-score (%)	97.5	91.7	91.6	94.5	93.8
AUC score	0.97	0.92	0.91	0.94	0.93

Recurrent Neural Networks (RNN): These networks are very effective at processing sequential data, which makes them valuable for identifying network traffic attacks. RNNs may, however, be less effective than more sophisticated deep learning models and are susceptible to vanishing gradient issues.

The INTRUMER model is an advanced intrusion detection solution that combines FOA for feature selection, HAT for advanced classification, and XPM for transparency. It guarantees higher accuracy in the detection process, effectively handling unbalanced datasets and maintaining an overall computational power efficiency. Through benchmarking on CICIDS 2017 and CSE-CIC-IDS 2018 datasets, INTRUMER depicted the best accuracy at 98.7% and F1-score at 97.8%, outperforming recent IDS methods in Table 14. In fact, FOA performs optimal feature selection and, hence, removes noise while enhancing classifier performance. HAT corrects highly complex hazards existing in network traffic. XPM guarantees explainability for real-world deployment. Compared to other models, such as transformer-based IDS by Hagar et al. [40] (F1-score: 96.2%) with high computational costs, and Khanday et al.'s [41] hybrid IDS (95.8% accuracy on TON_IoT), INTRUMER is scalable and suitable for distributed cloud environments. Rani et al.'s CNN-GRU IDS (F1-score: 94.1%) lacks feature selection, causing inconsistencies [42]. INTRUMER's optimized feature selection and efficiency ensure adaptability and effectiveness, setting a new standard in IDS design.

Table 14: Comparison of intrumer model based on existing model methodologies

Study	Year	Methodology	Dataset	Accuracy (%)	F1-score (%)	Limitations
Hagar et al.	2022	CNN + LSTM	CSE-CIC-IDS 2018	96.0	96.2	High computational cost
Khanday et al.	2023	Deep autoencoder + Random forest	TON_IoT	95.8	95.5	Lacks explainability
Rani et al.	2023	CNN + GRU-Based IDS	UNSW-NB15	94.5	94.1	No optimization for feature selection
Proposed INTRUMER	2024	Distributed HAT + FOA + NB + XPM	CICIDS 2017, NSL-KDD	98.7	97.8	Comprehensive and explainable

The accuracy of the INTRUMER model on the CICIDS 2017 dataset is 2.7% higher than that of Hagar et al., showing that it has a generalizability and effectiveness in a wide range of network setups. In addition, among the weaknesses of the Hagar et al., the computation overhead is significantly reduced when FOA is utilized as feature selection. Similarly, in Table 15 our methodology meets the important requirement of transparency in IDS/IPS solutions with the use of explainable outputs through XPM rather than Khanday et al. and Rani et al.'s [40–42].

Table 15: Comparison of INTRUMER model with metrices

Model	Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Hagar et al.	CSE-CIC-IDS 2018	96.0	95.8	95.6	96.2
Khanday et al.	TON_IoT	95.8	95.3	95.7	95.5
Rani et al.	UNSW-NB15	94.5	94.0	93.8	94.1
Proposed INTRUMER	CICIDS 2017	98.7	97.5	96.3	97.8

6 Discussion and Limitations

Compared to conventional techniques, the INTRUMER model is more resilient to a variety of attack patterns because it integrates the Falcon Optimization Algorithm (FOA) for feature selection and the Heterogeneous Attention Transformer (HAT) for categorization. A key component of real-time cybersecurity decision-making is the system's interpretability and transparency, which are improved by the Naïve Bayes (NB) algorithm.

Increase in Detection Accuracy: In terms of overall detection accuracy and classification precision, INTRUMER performs better than conventional models (such as RF and SVM). It performs better due to its attention-based feature selection and classification algorithms, which enable it to more successfully adjust to known and novel assault scenarios.

Managing Class Imbalance: Attacks (the minority class) are underrepresented in comparison to regular traffic, a problem that traditional IDS models frequently face. This is addressed by the FOA and HAT modules in INTRUMER, which enhance the model's capacity to learn from minority class data without overfitting, leading to increased precision and recall scores for DDoS and botnet attacks.

Processing Time & Efficiency: A comparison of the model's computational efficiency was also made. Despite their great accuracy, deep learning models such as CNN and RNN typically demand a lot more processing power. INTRUMER provides a superior option for real-time systems by balancing efficiency and performance.

7 Conclusion

In conclusion, the rapid proliferation of Cloud-based Internet of Things (CIoT) devices has increased the challenges of cybersecurity, which leads to unwanted network traffic and critical security concerns. These issues have become a pressing need, as traditional Intrusion Detection Systems (IDS) often struggle with unforeseen cyberattacks and the complexities of high-dimensional data. To address the identified challenges, this study develops INTRUMER, a novel distributed, explainable, heterogeneous transformer-based IDS with balanced accuracy, reliability, and security in CIoT. The proposed INTRUMER model involves the integration of advanced modules. Starting with the TC&TM module that employs the Falcon Optimization Algorithm (FOA) for feature selection and the NB algorithm for feature classification. The selected and categorized features are then passed through the Heterogeneous Attention Transformer module which uses contextual information to effectively interact with the features for accurate classification of network traffic as normal or malicious. Finally, the results are cleaned up by the Explainable Prevention Module which not only provides interpretability in classifier decisions but also generates emergency alarms to alert nearby IDS modules servers and Cloud-IoT devices. Extensive experiments using three benchmark IDS datasets CICIDS 2017, Honeypot, and NSL-KDD confirm the superior capability of INTRUMER in distinguishing different types of traffic with remarkable accuracy. Comparing the performance with state-of-the-art models further establishes the strength and effectiveness of INTRUMER as a very promising tool to improve CIoT security in response to evolving cyber threats.

Acknowledgement: None.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm their contribution to the paper as follows: study conception and design: Nazreen Banu A; data collection: Nazreen Banu A; analysis and interpretation of results: Nazreen Banu A, S.K.B. Sangeetha; review and editing: Nazreen Banu A, S.K.B. Sangeetha; original draft preparation: Nazreen Banu A and S.K.B. Sangeetha. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are openly available in the public repository at https://www.kaggle.com/datasets/chethuhn/network-intrusion-dataset, accessed on 04 December 2024.

Ethics Approval: Not applicable.

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

References

[1] N. Mishra and S. Pandya, "Internet of things applications, security challenges, attacks, intrusion detection, and future visions: A systematic review," *IEEE Access*, vol. 9, pp. 59353–59377, 2021. doi: 10.1109/ACCESS.2021.3073408.

[2] R. Zhao *et al.*, "A novel intrusion detection method based on lightweight neural network for internet of things," *IEEE Internet Things J.*, vol. 9, no. 12, pp. 9960–9972, 2021. doi: 10.1109/JIOT.2021.3119055.

[3] I. Ullah and Q. H. Mahmoud, "Design and development of a deep learning-based model for anomaly detection in IoT networks," *IEEE Access*, vol. 9, pp. 103906–103926, 2021. doi: 10.1109/ACCESS.2021.3094024.

[4] G. Abdelmoumin, D. B. Rawat, and A. Rahman, "On the performance of machine learning models for anomaly-based intelligent intrusion detection systems for the internet of things," *IEEE Internet Things J.*, vol. 9, no. 6, pp. 4280–4290, 2021. doi: 10.1109/JIOT.2021.3103829.

[5] M. Abdel-Basset, H. Hawash, R. K. Chakrabortty, and M. J. Ryan, "Semi-supervised spatiotemporal deep learning for intrusions detection in IoT networks," *IEEE Internet Things J.*, vol. 8, no. 15, pp. 12251–12265, 2021. doi: 10.1109/JIOT.2021.3060878.

[6] M. Ozkan-Okay, R. Samet, Ö. Aslan, and D. Gupta, "A comprehensive systematic literature review on intrusion detection systems," *IEEE Access*, vol. 9, pp. 157727–157760, 2021. doi: 10.1109/ACCESS.2021.3129336.

[7] N. R. Sai, G. S. C. Kumar, M. A. Safali, and B. S. Chandana, "Detection system for the network data security with a profound deep learning approach," in *2021 6th Int. Conf. Commun. Electron. Syst. (ICCES)*, IEEE, 2021, pp. 1026–1031.

[8] S. Ullah *et al.*, "TNN-IDS: Transformer neural network-based intrusion detection system for MQTT-enabled IoT networks," *Comput. Netw.*, vol. 237, no. 5, 2023, Art. no. 110072. doi: 10.1016/j.comnet.2023.110072.

[9] P. Singh and V. Ranga, "Attack and intrusion detection in cloud computing using an ensemble learning approach," *Int. J. Inf. Technol.*, vol. 13, no. 2, pp. 565–571, 2021. doi: 10.1007/s41870-020-00583-w.

[10] A. Aldallal and F. Alisa, "Effective intrusion detection system to secure data in cloud using machine learning," *Symmetry*, vol. 13, no. 12, 2021, Art. no. 2306. doi: 10.3390/sym13122306.

[11] V. Chang *et al.*, "A survey on intrusion detection systems for fog and cloud computing," *Future Internet*, vol. 14, no. 3, 2022, Art. no. 89. doi: 10.3390/fi14030089.

[12] Y. Aoudni *et al.*, "Cloud security based attack detection using transductive learning integrated with hidden markov model," *Pattern Recognit. Lett.*, vol. 157, no. 2, pp. 16–26, 2022. doi: 10.1016/j.patrec.2022.02.012.

[13] S. Einy, C. Oz, and Y. D. Navaei, "The anomaly-and signature-based IDS for network security using hybrid inference systems," *Math. Probl. Eng.*, vol. 2021, no. 1, 2021, Art. no. 6639714. doi: 10.1155/2021/6639714.

[14] S. Rajagopal, P. P. Kundapur, and K. S. Hareesha, "Towards effective network intrusion detection: From concept to creation on Azure cloud," *IEEE Access*, vol. 9, pp. 19723–19742, 2021. doi: 10.1109/ACCESS.2021.3054688.

[15] A. Thakkar and R. Lohiya, "A review on machine learning and deep learning perspectives of IDS for IoT: Recent updates, security issues, and challenges," *Arch. Comput. Methods Eng.*, vol. 28, no. 4, pp. 3211–3243, 2021. doi: 10.1007/s11831-020-09496-0.

[16] Z. K. Maseer, R. Yusof, N. Bahaman, S. A. Mostafa, and C. F. M. Foozy, “Benchmarking of machine learning for anomaly based intrusion detection systems in the CICIDS2017 dataset,” *IEEE Access*, vol. 9, pp. 22351–22370, 2021. doi: 10.1109/ACCESS.2021.3056614.

[17] Y. Otoum, D. Liu, and A. Nayak, “DL-IDS: A deep learning-based intrusion detection framework for securing IoT,” *Trans. Emerg. Telecomm. Technol.*, vol. 33, no. 3, 2022, Art. no. e3803. doi: 10.1002/ett.3803.

[18] M. A. Khan, “HCRNNIDS: Hybrid convolutional recurrent neural network-based network intrusion detection system,” *Processes*, vol. 9, no. 5, 2021, Art. no. 834. doi: 10.3390/pr9050834.

[19] P. R. Kanna and P. Santhi, “Unified deep learning approach for efficient intrusion detection system using integrated spatial-temporal features,” *Knowl. Based Syst.*, vol. 226, no. 1, 2021, Art. no. 107132. doi: 10.1016/j.knosys.2021.107132.

[20] Z. Wu, H. Zhang, P. Wang, and Z. Sun, “RTIDS: A robust transformer-based approach for intrusion detection system,” *IEEE Access*, vol. 10, no. 3, pp. 64375–64387, 2022. doi: 10.1109/ACCESS.2022.3182333.

[21] S. Ho, S. Al Jufout, K. Dajani, and M. Mozumdar, “A novel intrusion detection model for detecting known and innovative cyberattacks using convolutional neural network,” *IEEE Open J. Comput. Soc.*, vol. 2, pp. 14–25, 2021. doi: 10.1109/OJCS.2021.3050917.

[22] F. Ullah, S. Ullah, G. Srivastava, and J. C. W. Lin, “IDS-INT: Intrusion detection system using transformer-based transfer learning for imbalanced network traffic,” *Digit. Commun. Netw.*, vol. 10, no. 1, pp. 190–204, 2024. doi: 10.1016/j.dcan.2023.03.008.

[23] G. S. Lalotra, V. Kumar, A. Bhatt, T. Chen, and M. Mahmud, “iReTADS: An intelligent real-time anomaly detection system for cloud communications using temporal data summarization and neural network,” *Secur. Commun. Netw.*, vol. 2022, no. 1, 2022, Art. no. 9149164. doi: 10.1155/2022/9149164.

[24] W. Elmasry, A. Akbulut, and A. H. Zaim, “A design of an integrated cloud-based intrusion detection system with third party cloud service,” *Open Comput. Sci.*, vol. 11, no. 1, pp. 365–379, 2021. doi: 10.1515/comp-2020-0214.

[25] S. Huang *et al.*, “HitAnomaly: Hierarchical transformers for anomaly detection in system log,” *IEEE Trans. Netw. Serv. Manag.*, vol. 17, no. 4, pp. 2064–2076, 2020. doi: 10.1109/TNSM.2020.3034647.

[26] Q. Ma, C. Sun, B. Cui, and X. Jin, “A novel model for anomaly detection in network traffic based on kernel support vector machine,” *Comput. Secur.*, vol. 104, no. 2, 2021, Art. no. 102215. doi: 10.1016/j.cose.2021.102215.

[27] L. Nie *et al.*, “Intrusion detection in green internet of things: A deep deterministic policy gradient-based algorithm,” *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 2, pp. 778–788, 2021. doi: 10.1109/TGCN.2021.3073714.

[28] L. Zou, X. Luo, Y. Zhang, X. Yang, and X. Wang, “HC-DTTSVM: A network intrusion detection method based on decision tree twin support vector machine and hierarchical clustering,” *IEEE Access*, vol. 11, pp. 21404–21416, 2023. doi: 10.1109/ACCESS.2023.3251354.

[29] A. Ponmalar and V. Dhanakoti, “An intrusion detection approach using ensemble support vector machine based chaos game optimization algorithm in big data platform,” *Appl. Soft Comput.*, vol. 116, no. 7, 2022, Art. no. 108295. doi: 10.1016/j.asoc.2021.108295.

[30] E. Mushtaq, A. Zameer, M. Umer, and A. A. Abbasi, “A two-stage intrusion detection system with auto-encoder and LSTMs,” *Appl. Soft Comput.*, vol. 121, no. 6, 2022, Art. no. 108768. doi: 10.1016/j.asoc.2022.108768.

[31] S. Krishnaveni, S. Sivamohan, S. S. Sridhar, and S. Prabakaran, “Efficient feature selection and classification through ensemble method for network intrusion detection on cloud computing,” *Cluster Comput.*, vol. 24, no. 3, pp. 1761–1779, 2021. doi: 10.1007/s10586-020-03222-y.

[32] R. SaiSindhuTheja and G. K. Shyam, “An efficient metaheuristic algorithm based feature selection and recurrent neural network for DoS attack detection in cloud computing environment,” *Appl. Soft Comput.*, vol. 100, no. 1, 2021, Art. no. 106997. doi: 10.1016/j.asoc.2020.106997.

[33] A. Javadpour, P. Pinto, F. Ja’fari, and W. Zhang, “DMAIDPS: A distributed multi-agent intrusion detection and prevention system for cloud IoT environments,” *Cluster Comput.*, vol. 26, no. 1, pp. 367–384, 2023. doi: 10.1007/s10586-022-03621-3.

[34] O. D. Okey, D. C. Melgarejo, M. Saadi, R. L. Rosa, J. H. Kleinschmidt and D. Z. Rodríguez, "Transfer learning approach to IDS on cloud IoT devices using optimized CNN," *IEEE Access*, vol. 11, no. 4, pp. 1023–1038, 2023. doi: 10.1109/ACCESS.2022.3233775.

[35] A. K. Sangaiah, A. Javadpour, F. Ja'fari, P. Pinto, W. Zhang and S. Balasubramanian, "A hybrid heuristics artificial intelligence feature selection for intrusion detection classifiers in cloud of things," *Cluster Comput.*, vol. 26, no. 1, pp. 599–612, 2023. doi: 10.1007/s10586-022-03629-9.

[36] M. C. Gaitan-Cardenas, M. Abdelsalam, and K. Roy, "Explainable AI-based intrusion detection systems for cloud and IoT," in *2023 32nd Int. Conf. Comput. Commun. Netw. (ICCCN)*, IEEE, 2023, pp. 1–7.

[37] S. M. T. Nizamudeen, "Intelligent intrusion detection framework for multi-clouds-IoT environment using swarm-based deep learning classifier," *J. Cloud Comput.*, vol. 12, no. 1, pp. 134, 2023. doi: 10.1186/s13677-023-00509-4.

[38] M. Arunkumar and K. A. Kumar, "GOSVM: Gannet optimization based support vector machine for malicious attack detection in cloud environment," *Int. J. Inf. Technol.*, vol. 15, no. 3, pp. 1653–1660, 2023. doi: 10.1007/s41870-023-01192-z.

[39] A. M. Ali, F. Alqurashi, F. J. Alsolami, and S. Qaiyum, "A double-layer indemnity enhancement using LSTM and HASH function technique for intrusion detection system," *Mathematics*, vol. 11, no. 18, 2023, Art. no. 3864.

[40] A. A. Hagar and B. W. Gawali, "Deep learning for improving attack detection system using CSE-CICIDS2018," *NeuroQuantology*, vol. 20, no. 6, 2022, Art. no. 3064.

[41] S. A. Khanday, H. Fatima, and N. Rakesh, "Implementation of intrusion detection model for DDoS attacks in lightweight IoT networks," *Expert. Syst. Appl.*, vol. 215, no. 8, 2023, Art. no. 119330. doi: 10.1016/j.eswa.2022.119330.

[42] S. Rani and S. Kumar, "A comprehensive analysis of intrusion detection datasets: Evaluation, challenges, and insights," in *2023 Seventh Int. Conf. Image Inf. Process. (ICIIP)*, IEEE, 2023, pp. 547–551.