Neural Network-Powered Intrusion Detection in Multi-Cloud and Fog Environments

Yanfeng ZHANG, Zhe XU
College of Artificial Intelligence, Jiaozuo University, Jiaozuo, Henan, 454000, China

***Abstract*—Cloud Computing has revolutionized the technological landscape, offering a platform for resource provisioning where organizations can access computing resources, storage, applications, and services. The shared nature of these resources introduces complexities in ensuring security and privacy. With the advent of edge and fog computing alongside cloud technologies, the processing, data storage, and management paradigm faces challenges in safeguarding against potential intrusions. Attacks on fog computing, IoT cloud, and related advancements can have pervasive and detrimental consequences. To address these concerns, various security standards and schemes have been suggested and deployed to enhance fog computing security. In particular, the focus of these security measures has become vital due to the involvement of multiple networks and numerous fog nodes through which end-users interact. These nodes facilitate the transfer of sensitive information, amplifying privacy concerns. This paper proposes a multi-layered intermittent neural network model tailored specifically for enhancing security in fog computing, especially in proximity to end-users and IoT devices. Emphasizing the need to mitigate privacy risks inherent in extensive network connections, the model leverages a customized adaptation of the NSLKDD dataset, a challenging dataset commonly applied to evaluate intrusion detection systems. A range of current models and feature sets are rigorously investigated to quantify the effectiveness of the proposed approach. Through comprehensive research findings and replication studies, the paper demonstrates the stability and robustness of the suggested method versus various performance metrics employed for intrusion detection. The assessment illustrates the model's superior capability in addressing privacy and security challenges in hybrid cloud environments incorporating intrusion detection systems, offering a promising solution for the evolving landscape of cloud-based computing technologies.**

***Keywords*—Cloud computing; fog computing; intrusion detection; privacy protection; neural network**

I. Introduction

Cloud computing has gained prominence as a prominent technology within the realm of Information Technology (IT) in recent years. The inception of cloud computing can be traced back to 2006, when Google introduced this groundbreaking concept. Subsequently, with the evolution of computer technology and novel communication paradigms, the IT landscape witnessed a rapid transformation, elevating the significance of this innovation for both individuals and organizations within the industry [1]. Cloud computing has revolutionized the accessibility and management of computing resources, enabling organizations to leverage shared services, applications, and data storage through remote servers. The evolution of cloud technology has expanded to incorporate edge and fog computing paradigms, emphasizing decentralized data processing and analytics closer to the data source [2]. However, this expansion brings forth a myriad of security and privacy challenges. Particularly, the intersection of multi-cloud environments and the criticality of intrusion detection in fog computing is becoming increasingly complex and crucial in safeguarding sensitive data and preventing unauthorized access [3].

The essence of cloud computing lies in its shared infrastructure, allowing multiple users to access resources and services remotely. However, with the advent of multi-cloud architectures, organizations employ services from different cloud providers, leading to interconnectivity complexities [4]. Multi-cloud setups aim to enhance performance, reduce latency, and mitigate risks associated with a single-cloud dependency. Nevertheless, integrating multiple clouds amplifies security vulnerabilities, requiring robust intrusion detection systems to counter potential threats and breaches. Edge and fog computing have emerged as pivotal components in the cloud ecosystem, focusing on processing data near the data source to reduce latency and enhance efficiency. This proximity to end-users and IoT devices in fog computing introduces a new set of security challenges, especially regarding privacy concerns and intrusion risks. The transfer of sensitive data across numerous fog nodes poses a significant threat, necessitating sophisticated security measures that can protect privacy and detect intrusions effectively in these intricate network architectures.

This paper proposes an Intrusion Detection System (IDS) leveraging neural network technologies specifically tailored for fog computing to deal with the security and privacy threats in multi-cloud environments. The application of neural networks in intrusion detection aims to fortify security measures, providing a more adaptive and sophisticated approach to identifying and mitigating potential threats in multi-cloud and fog environments. This research explores developing and evaluating a novel IDS framework to tackle the escalating security challenges arising from the interconnection of multi-cloud infrastructures, thereby aiming to ensure data integrity and user privacy in fog computing setups.

II. Related Work

Within the domain of cloud, IoT, and interconnected computing environments, ensuring robust security against intrusions is of utmost importance. This section comprehensively explores and compares various cutting-edge methodologies and approaches employed for IDSs. Each study's distinct methodology, datasets utilized, performance metrics,

and principal findings are evaluated to provide a comprehensive understanding of their efficacy for enhancing security measures and minimizing potential threats within these dynamic technological landscapes. Table I provides a concise comparative overview of various intrusion detection methodologies and their respective outcomes.

The advent of the IoT has facilitated extensive connectivity across many objects and services, leading to a susceptibility to IoT and cloud malware infections. Consequently, cybersecurity stands as a crucial concern in establishing resilient IoT systems [14]. Abd Elaziz, et al. [5] have capitalized on recent advancements in swarm intelligence approaches and the progress in deep neural networks to develop an effective IDS for cloud- and IoT-based scenarios. Initially, deep neural networks extract valuable information from IDS data. Subsequently, a proficient feature selection method is introduced, leveraging the Capuchin Search Algorithm (CapSA), a recently proposed swarm intelligence optimizer [6]. The resultant model, termed CNN-CapSA, is rigorously tested using four distinct datasets, specifically CIC2017, KDD99, BoT-IoT, and NSLKDD. Furthermore, comprehensive empirical comparisons are conducted against alternative optimization methods, encompassing various criteria for classifying results. The findings substantiate that the proposed method performs well across all analyzed datasets.

With growing internet traffic and the rise of attacks against the cloud ecosystem, intrusion monitoring is becoming more complicated. An attacker may gain access to a variety of protocol interfaces, such as Hypertext Transfer Protocol (HTTP), Domain Name System (DNS), and Message Queue Telemetry Transport (MQTT), leading to data breaches and security vulnerabilities. Traditional machine learning algorithms like neural networks, fuzzy logic, and support vector machines have been commonly employed in IDSs. However, these methods exhibit limitations such as slow convergence, inaccurate results, vanishing gradients, excessive fitting, and subpar efficiency. To address these challenges, Geetha and Deepa [7] have introduced a novel approach, the Fisher kernel-based PCA dimensionality reduction algorithm in conjunction with a grey wolf optimizer based weight dropped BiLSTM classifier (FKPCA-GWO WDBiLSTM) to detect intrusions. The PCA algorithm is initially applied with data records, utilizing the Fisher kernel and Fisher score to separate dimensions linearly. Subsequently, the WDBiLSTM structure captures persistent dependencies and extracts features bidirectionally. The GWO optimizes the recurrent weights, ensuring accurate classification and distinguishing between normal and attack instances. The proposed mechanism has been rigorously evaluated on four datasets. The findings demonstrate superior F-measure, specificity, sensitivity, precision, and accuracy performance compared to previous approaches such as FCM-SVM, DRIOTIDS, BiCIDS, and Fuzzy-SMO.

Conventional network IDSs cannot adequately fulfill the security requisites of IoT deployments. Addressing this limitation, Lin, et al. [8] have integrated machine learning and cloud computing into IoT IDS to enhance its detection capabilities. Typically, conventional IDSs demand substantial training duration and are unsuitable for cloud computing owing to cloud nodes' restricted storage and computing capabilities. Hence, there is a pressing need to investigate IDSs characterized by lightweight, superior detection accuracy, and swift training time. Selecting a suitable classification methodology is crucial when implementing cloud-based IDSs and is essential for an effective defense response to intrusions while mitigating intrusions. The authors extensively discussed issues concerning IoT intrusion mitigation in cloud computing contexts. They employed the Multi-Feature Extraction Extreme Learning Machine (MFE-ELM) algorithm, introducing a multi-feature extraction procedure within cloud servers. Afterward, MFE-ELM was used in cloud servers to identify cybersecurity breaches. Several tests utilized a classical dataset, involving stages including data preprocessing, designing features, training the model, and data analysis. The simulation outcomes demonstrated the suggested algorithm's effectiveness in detecting a substantial percentage of network data packets, exhibiting commendable results. It also proved adept at efficiently detecting intrusions into heterogeneous IoT data from cloud nodes. Moreover, the algorithm facilitates real-time identification of nodes posing severe security threats within the cloud cluster, enabling the cloud server to take immediate security measures.

The surge in cloud computing has raised persistent concerns about privacy and security. Addressing these issues, Al-Ghuwairi, et al. [9] have introduced a new method aimed at immediately identifying malicious activities in cloud computing through time series analysis. This innovative technique integrates feature selection methods with a predictive technique derived from the Facebook Prophet system to determine its effectiveness. The feature selection process combines historical data analysis with anomaly detection, stationarity, and correlation analyses to resolve the complexities of identifying relationships among time series variations and potential threats. This approach significantly reduces the number of predictors used in the predictive model while optimizing various parameters like Dynamic Time Warping (DTW), Median Absolute Percentage Error (MdAPE), Mean Absolute Percentage Error (MAPE), Root Mean Squared Error (RMSE), Mean Squared Error (MSE), and Mean Absolute Error (MAE). It has also considerably minimized cross-validation, prediction, and training times. Although memory consumption is stable, utilization time dropped significantly, leading to a significant decline in resource usage. This study offers a unified approach to effective intrusion detection in cloud computing by exploiting time series anomalies, using a collaborative feature selection process and the Facebook Prophet prediction engine. The results underscore the improved performance and efficiency achieved by this approach, enhancing the progress of intrusion detection strategies in cloud computing security.

In IoT environments, foundational to computing services, vulnerabilities and cyber threats remain constant concerns. Adversaries continuously seek weak points within these computing environments to perpetrate damage, posing intricate challenges. Consequently, employing intrusion prevention and detection solutions becomes essential for securing IoT environments. However, recent strategies in this domain encounter limitations, notably the inability to detect unknown attacks and susceptibility to single points of failure. To address these constraints, Javadpour, et al. [10] have introduced a novel

approach: a distributed multi-agent IDS, effectively mitigating these issues. It uses a six-stage detection procedure to categorize network activities as safe or dangerous. The suggested method was validated using the NSLKDD and KDD Cup 99 datasets. Test outcomes were evaluated against other methodologies in terms of f-score, accuracy, and recall metrics.

To deal with the issue of low accuracy in conventional tracking signal detection algorithms within the traditional cloud-side collaborative computing setting, Zhong and Zhong [11] have proposed a novel deep learning-based track signal intrusion detection method within the cloud-edge collaborative computing setup. The approach involves constructing the core framework of the IDS by holistically examining core networks, communication links, and infrastructure and integrating edge computing into cloud services. The proposed method leverages CNN-attention-based BiLSTM (Convolutional Neural Networks-attention-based Bi-directional Long Short-Term Memory) as a central layer of the method in order to train on historical datasets, thereby presenting a deep learning-based technique. Additionally, dropout and pooling layers are incorporated to avoid overfitting and enhance track signal intrusion detection. The pooling layer is integrated to accelerate model convergence, eliminate redundant features, and diminish feature dimensionality, while the dropout layer aims to prevent overfitting. The proposed IDS is compared and analyzed using simulation experiments against three other methods under identical conditions. Results indicate that the proposed method has a higher F1 value than the other techniques across four sample datasets. The F1 value varies from 0.94 to 0.96, demonstrating superior performance to other comparison algorithms. This method proves crucial for resolving IDS signal concerns within the cloud-edge cooperative setting and lays the conceptual foundation to track signal IDS direction.

Implementing an anomaly-based IDS is key to maintaining the integrity of database records by identifying and isolating anomalies, particularly when unexpected changes are detected. In advanced networking environments, classification and clustering methods based on machine learning serve as an effective approach to identifying and categorizing anomalous IDS attacks. Machine learning is a swift, cost-effective, and flexible tool for constructing intrusion detection schemes capable of addressing a wide range of threats. Samunnisa, et al. [12] have introduced a proficient hybrid clustering and classification model for implementing an anomaly-based IDS, particularly for classifying malicious attack types such as normal (no intrusion), Denial of Service (DoS), Probe, User to Root (U2R), and Remote to Local (R2L) attacks. This approach utilizes threshold-based functions and is tested using two different threshold values, specifically 0.01 and 0.5, across the NSLKDD and KDDcup99 datasets. Performance evaluation metrics such as Detection Rate, False Alarm Ratio, and Accuracy have been employed to assess the effectiveness of the proposed methodology. Results showcase that applying the proposed approach, particularly the K-means combined with Random Forest at two distinct threshold values, exhibits superior classification accuracy. Specifically, it achieved a detection rate, false alarm rate, and accuracy of 99.8%, 99.7%, and 0.1%, respectively, on the NSLKDD dataset and 98.2%, 98.1%, and 2% on the KDDcup99 dataset.

TABLE I. OVERVIEW OF PREVIOUS IDS METHODOLOGIES

Study	Methodology	Datasets used	Performance metrics
Abd Elaziz, et al. [5]	CNN-CapSA utilizing deep neural networks for IDS in cloud- and IoT-based scenarios	CIC2017, KDD99, BoT-IoT, and NSLKDD	Comparative analysis against alternative optimization methods
Geetha and Deepa [7]	FKPCA-GWO WDBiLSTM for intrusion detection	Four datasets	F-measure, specificity, sensitivity, precision, and accuracy
Lin, et al. [8]	MFE-ELM algorithm for IoT IDS utilizing cloud computing	Classical dataset	Model performance assessment
Al-Ghuwairi, et al. [9]	Early intrusion detection in cloud computing using time series data	Time series data	Performance metrics (MAE, MSE, RMSE, etc.)
Javadpour, et al. [10]	Distributed multi-agent IDPS (DMAIDPS) for IoT environments	KDD Cup 99 and NSLKDD	Recall, accuracy, and F-score
Zhong and Zhong [11]	Deep learning-based track signal intrusion detection in cloud-edge computing	Simulation experiments	F1 value comparison
Samunnisa, et al. [12]	Hybrid clustering and classification model for anomaly-based IDS	NSLKDD and KDDcup99	Detection rate, false alarm ratio, and accuracy

As reviewed literature, the current research in the field of intrusion detection systems (IDSs) lacks a comprehensive understanding of the efficacy of various methodologies and approaches, particularly in the context of cloud, IoT, and interconnected computing environments. Existing studies often focus on specific techniques without providing a comparative analysis of their performance across different datasets and scenarios. Additionally, there is a need for innovative solutions that address the evolving cybersecurity threats posed by cloud and IoT malware infections, as well as the challenges associated with integrating IDSs into these dynamic technological landscapes. Furthermore, the scalability and efficiency of IDSs in handling growing internet traffic and complex network protocols remain understudied areas. Moreover, conventional IDSs may not adequately fulfill the security requirements of IoT deployments, necessitating the development of lightweight and efficient detection mechanisms tailored for IoT environments. Finally, the effectiveness of anomaly-based IDSs in maintaining database integrity and identifying novel attack types in advanced networking environments requires further exploration and validation.

III. PROPOSED METHOD

The cloud represents a significant asset for IoT environments, offering a comprehensive solution to various IoT challenges. However, integrating cloud technology introduces several challenges, encompassing security and privacy concerns, latency, integrity, and bandwidth limitations. IDSs typically operate in non-cloud environments based on a trust-based cooperative model. Researchers have proposed a trust-based cooperative IDS that operates through collaboration

among local IDS units, identifying new attacks unknown to other IDSs. These systems utilize data from diverse IDSs to facilitate intrusion detection [15]. A key architectural aspect of these cooperative IDSs involves a feedback mechanism for reliable data collection. An incentivized communication model also encourages IDS nodes to share inputs among known nodes to prevent malicious activities. However, limitations exist within the current trust-based cooperative IDS framework, particularly in soliciting input from numerous IDSs. The proposed algorithm, based on relevant theoretical concepts, allows a collective group of IDSs to establish their collaboration in a manner that enhances their detection accuracy, even in the presence of untrusted IDS units. Notably, existing cooperative IDSs encounter considerable delays primarily due to the algorithmic complexity associated with employing comprehensive algorithms.

The overall strategy requires significant computational time, contingent on multiple factors, such as the consulted IDS, the expertise of the IDSs, and various trust levels. Uncertainties surrounding the receipt of inputs across various levels, especially within IDS associations, internet speeds, and other ambiguous elements, lead to potential delays in decisions about alerting potential threats due to missing input from individual IDS. Consequently, cooperative IDS decisions are not feasible in time-sensitive settings. In the initial phase, a real-world IoT-based smart home prototype was built, and the regular activities of every device within the IoT network were monitored. Subsequently, malicious tests were conducted, inducing anomalous network traffic to these devices. These stages facilitated the application of an Artificial Intelligence and deep learning based approaches using well-prepared training data, forming the basis of the intrusion detection model [16]. The developed system exhibited superior detection accuracy within an acceptable time frame. A logarithmic minimal density ratio adjustment was applied to the NSLKDD dataset features to achieve enhanced detection capabilities to produce higher-quality, representative features. Employing SVM to perform classification, the experimental findings revealed high accuracy and detection rates. Fig. 1 illustrates the structure of the proposed cooperative IDS, comprising six cloud providers.

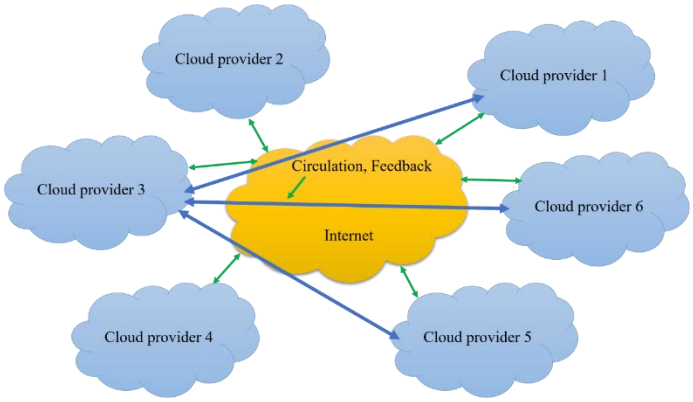

Fig. 1. Cooperative IDS architecture.

The suggested conceptual framework for IoT security is thoroughly examined, outlining an intrusion detection scheme composed of two primary processors: one for classification and another for traffic analysis. Traffic association logs follow processing within traffic handling units, resulting in data suitable for deep neural network processing by the classification engines, categorizing these associations as normal [13]. The model is deployable in fog computing, closely situated near IoT devices and end users. It incorporates a recurrent neural network based on a modified variant of the backpropagation procedure to enhance the predictive capacity of regular/threat identification. A recursive process within the network, wherein non-linear components' outputs are transformed into linear components, ensures rapid reaction and dependable continuous security for the IoT network. This recursive architecture serves as the core engine for classification-based traffic analysis. Fig. 2 illustrates the overall architecture of the developed conceptual framework for IoT security.

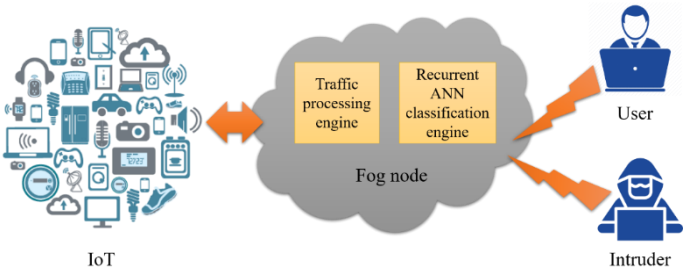

Fig. 2. The conceptual framework of IoT security.

The traffic handling engine employs the NSLKDD dataset to train, test, and validate models. This dataset comprises information that characterizes the network traffic of the networking system, often exhibiting inconsistencies. The pre-treatment of acquired traffic data becomes a crucial filter for the classification engine. Raw traffic data is preprocessed in four key steps within traffic preprocessors. These steps encompass symbolic-to-numerical conversion, data feature reduction, min-max standardization, and data sampling. The symbolic-to-numerical mapping and label representation are visually represented, facilitating the conversion of representative values (properties) of the NSLKDD dataset into numerical values. Flag features are denoted as {pstr = 4, …, s2 = 14}0, service features such as private = {private = 16, Netsat = 20}, and protocol features are denoted as Protocol = (tcp = 1, udp = 2, icmp = 3}. Numeric values for each characteristic are assigned based on the frequency of occurrence. As the frequency increases, the corresponding numerical value decreases, ensuring that attributes with the least frequency are not overshadowed by attributes with the highest frequency values. Fig. 3 displays an overview of the NSLKDD dataset. The different subclasses of attacks are encapsulated and categorized into their main classes as the last step of dataset coding. Table II offers classification details for the NSLKDD dataset.

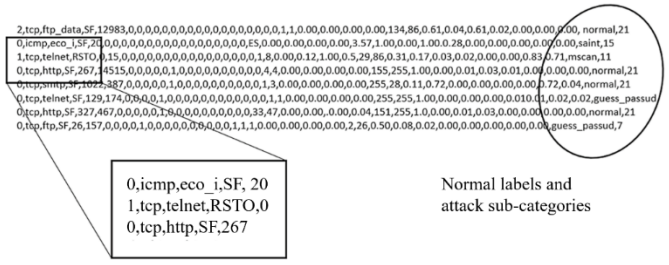

Fig. 3. NSLKDD dataset.

TABLE II. CLASSIFICATION OF THE NSLKDD DATASET

Classes	Sub-classes	Numeric code
U2R	Buffer flow, Xterm, Ps, Perl, load module	1
R2L	Warezmater, multi-hop, PHF, ftp-write, and guess password	2
Probe	Mscan, Nmap, Ipswd, and Satan	3
DoS	Worm, Mailbomb, Teardrop, Smurf, Neptune, Back, and Land	4

Table II presents two types of engine-parallel encapsulations, where all training dataset records are encapsulated into normal. This table lists 40 attack names encapsulated into four significant classifications. In the feature reduction process, all consistent-valued attributes that do not impact the analytical outcomes post-neural network analysis are eliminated from all records in the traffic data. Within this work, features have been removed where zero values reduce the data volume from 41 to 25 attributes. To ensure the traffic data values fall within a standardized range appropriate for neural network inputs, data values are scaled for min-max normalization. Direct data transformation involving min-max standardization has been employed in this work to achieve this aim.

$$S' = \frac{(S - min_f)}{(max_f - min_f)}(\quad max'_f - \quad min'_f) + (\quad min'_f) \tag{1}$$

Due to the infrequent occurrence of R2L and U2R attacks, the neural network tends to identify them as noisy signals, given their minimal impact on weight updating. Consequently, this leads to a significant weakness in detecting these particular attacks. To address this issue, both U2R and R2L attacks are multiplied by incorporating numerous instances of these attacks into various data points. This oversampling process generates new instances and a more widespread representation of these rare attack types. The proposed intrusion detection engine consists of two distinct detection stages with two deep recursive neural networks that have different internal structures, configuration parameters, and hyperparameters. The primary layer focuses on detecting DoS attempts, known as one of the primary threats that disrupt IoT systems, in addition to identifying other attack types. For heightened security measures, the output from the primary layer is further filtered by a secondary layer, featuring a different internal structure, configuration parameters, and selection criteria specifically tuned to detect attacks overlooked by the base layer, especially U2R and R2L attacks. To enable accurate detection, the second layer was trained using a dataset derived from the primary layer, excluding the DoS attacks.

Fig. 4 depicts a block diagram of the IDS system employing the creation of non-linear embeddings from the previous state through a deep recursive structure. The *h (t - 1)* represents the previous state, while *h(t)* represents the current state, including the incorporation of recursive gain. Traditional backpropagation encountered a gradient problem within the conventional network structure. The problem is mitigated by introducing a feedback mechanism that connects the prior state with the present state, elevating the current state. The proposed model is decomposed into four key steps: backpropagation to the hidden layer, backpropagation to the output layer, weight adjustment, and feedback propagation. An Artificial Neural Network (ANN) comprises basic computational elements called neurons, interconnected by weights. The structure of neurons is layered, ensuring complete connectivity between the preceding and subsequent layers.

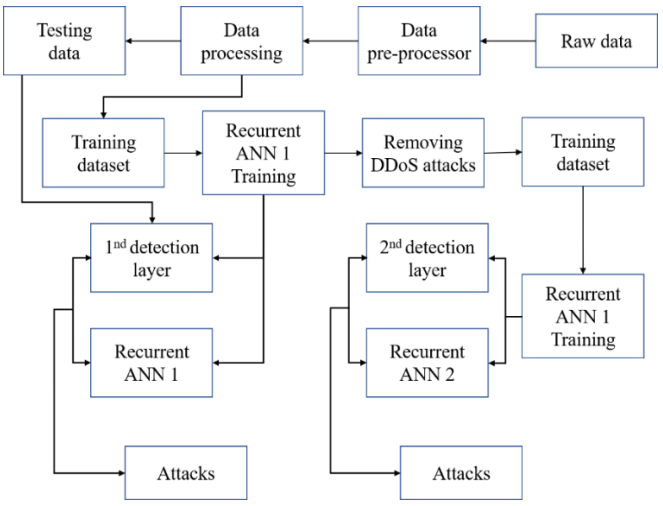

Fig. 4. Block diagram of the proposed IDS model.

Feature selection is crucial in eliminating irrelevant features to enhance the model's performance. The primary aim of feature selection is to identify a subset of features that result in higher classification accuracy. The model employs the gain rate ranking-based feature selection scheme, which overcomes biases and aids in feature subset selection by considering split data and normalizing information gain accordingly. The attribute selected with the maximum gain ratio is the splitting attribute. However, to prevent instability due to split information, a constraint is introduced, mandating that each test must gain more information than the average gain of all tests. A three-layered system governed by feedforward computation aims to streamline complexity and enhance training algorithms. These layers consist of the input, hidden, and output layers. The primary goal of training is to optimize network parameters to enable effective classification. In a network with an input vector for configuration, hidden nodes, and output nodes, connections between the input layer and hidden nodes are known as "weights." Similar connections between hidden nodes and output units are also named accordingly. The weighted response of a given sample ξ is computed back to the input summation unit, executed as an added weighted edge. Considering the bias, the length of the input vector is expanded by two layers, encompassing bias and the weighted edge, affecting all layers. Eq. (2) presents the linear output from the linear part, while Eq. (3) demonstrates output from the hidden layer, utilizing a sigmoidal function as a transfer function. These equations can be extended and applied to networks with multiple layers. O_{jss} denotes the hidden layer output, and O_{lin} represents the linear output.

$$O_{lin} = \sum_{t=1}^{n+2} W_{i,j}\, O_i \tag{2}$$

$$O_{jss} = f\left[\sum_{t=1}^{n+1} W_{i,j}\, O_i\right] \tag{3}$$

During the backpropagation process in the output layer, the primary objective is to determine partial derivatives of the error

signal *Er* concerning $W_{i,j}$. Error signals are a fundamental component in all typical versions of backpropagation neural networks. The summation outputs are obtained, and this difference measures the desired outputs.

$$E_{rj} = \sum_{p=1}^{p} (o_j^p - o_j^{Ep})^2 \quad (4)$$

In Eq. (4), E_{rj} represents the error signal of the j^{th} neuron in the output layer. o_j^p signifies the desired output of the j^{th} neuron for the p^{th} pattern, o_j^{Ep} while indicating a possible output of the j^{th} neuron for the p^{th} design; here, p denotes the pattern or data instance. Specifically, when mentioning $p = 1$, it refers to the representation at pattern 1. This training method encountered a problem where the estimation of one-layer output relied on the information received from the previous layer. At the start of the training, the previous layer remained untrained, leading to inaccurate estimations.

IV. Results and Performance Evaluation

An in-depth examination of the results and discussions from the experimentation is presented in this section. We tested our intrusion detection model with different operational configurations and compared the results to previous research findings. An Intel CoreTM i7 3.2 processor and 16 GB RAM running on Windows 8 were utilized within the MATLAB 2020b environment to develop the proposed IDS model. While the KDD-Cup-99 dataset is commonly employed for such purposes, a substantial redundancy within this dataset presents a significant challenge. Due to the high repetition of records, many AI-based IDSs trained on the KDD-Cup-99 dataset produced exceptional results across various evaluation metrics without significant compromises or integrated tuning procedures. Thus, we decided to employ the KDD-Cup-99 dataset as the primary source to compare various AI models according to their detection performance. Analytical systems discovered that these redundant records were being used. To address this, these systems were validated and examined using redundant records, enhancing inaccurate and inconsistent detection performance. While the NSLKDD collection was chosen to overcome the redundancy in the KDD-Cup-99 collection, it resolved the imbalance resulting from highly and less frequent U2R and R2L attacks. Our work employed an oversampling technique, as previously mentioned in the outlined scheme. Within the multiple-layer structure, the confusion matrix is the cornerstone of every performance measurement, constructed separately. It contains essential output class information. Key metrics within the confusion matrix include True, False Negative (FN), False Positive (FP), and True Negative (TN).

A value denoting normal instances within a dataset is expressed as true. TN refers to the correct identification of normal instances correctly. On the other hand, FP and FN indicate misclassifications in the classification results. When attack records are incorrectly labeled as normal instances, it results in a False Positive, posing a significant issue for the privacy and accessibility of organizational resources as attackers often bypass intrusion detection systems. Conversely, a False Negative occurs when instances of attacks are incorrectly labeled as normal. An FP essentially indicates appropriate behavior, commonly recognized as a false alert rate in intrusion detection scenarios.

$$Pr\,e\,cision = \frac{(True\ Positive)}{(True\ Positive + False\ Positive)} \quad (5)$$

$$Pr\,e = \frac{(TP)}{(TP+FP)} \quad (6)$$

$$Accuracy = \frac{(True\ Positive)}{(True\ Positive+True\ Negative+False\ Positive+False\ Negative)} \quad (7)$$

$$Acc = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad (8)$$

$$Detection\ rate = \frac{(True\ Positive)}{(True\ Positive+False\ Negative)} \quad (9)$$

$$DR = \frac{(TP)}{(TP+FN)}\lambda \quad (10)$$

$$False\ Positive\ Rate = \frac{(False\ Positive)}{(False\ Negative+True\ Negative)} \quad (11)$$

$$FPR = \frac{(FP)}{(FP+TN)} \quad (12)$$

$$False\ Negative\ Rate = \frac{(False\ Negative)}{(False\ Negative+True\ Positive)} \quad (13)$$

$$FPR = \frac{(FN)}{(FN+TP)} \quad (14)$$

$$F_1 - Measure = \frac{2}{\frac{1}{Detectionrate}+\frac{1}{Pr\,ecision}} \quad (15)$$

Eq. (16) yields the Matthews Correlation Coefficient (MCC), the phi coefficient, based on the aforementioned equations.

$$phi = \frac{(TP \times TN - FP \times FN)}{\sqrt{(TP+FN)(TP+FP)(TN+FN)(TN+FN)}} \quad (16)$$

While many performance assessments typically focus on detection rate and accuracy within the proposed model, two new performance metrics have been introduced: MCC and Kappa. The primary reason behind this additional variation is to measure recursive network performance robustness. The MCC value varies from -1 to 1. In predictive modeling, performance metrics alone may not comprehensively depict the classification, particularly in highly imbalanced datasets. Sixty-eight thousand training records were employed as input to the first layer simulation, and 40,000 records were utilized as test data. Performance measurements are presented in Table III.

TABLE III. Performance Results

Detection layers	Metrics				
	CCM	***FP rate***	***Detection rate***	***Precision***	***Accuracy***
First layer	0.87%	9.9%	96.6%	90.1%	91.5%
Second layer	0.93%	9.3%	95.2%	91.3%	93.7%

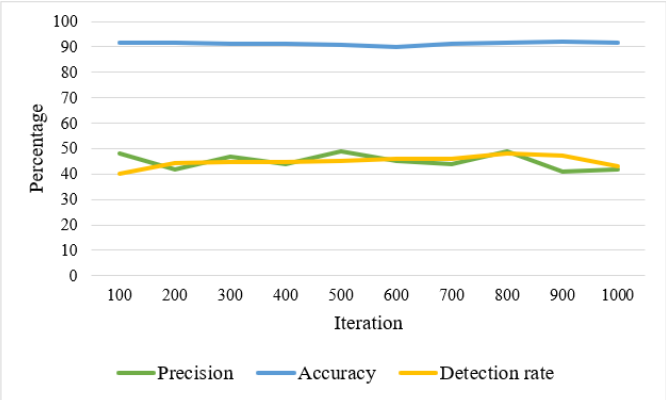

Fig. 5. Performance improvement vs. iteration.

Fig. 5 illustrates model performance evaluation concerning detection accuracy, precision, and rate. The graph demonstrates that precision, accuracy, and detection rate change as the number of iterations increases. A detection accuracy of 91.9% is observed at iteration 500. Considering the recursive nature of the organization, it remains unclear whether the number of iterations during the training stage influences the detection of unusual and hard-to-detect intrusions. Fig. 6 illustrates how recursive gain affects model performance. With increasing recursive gain, detection rates tend to decrease.

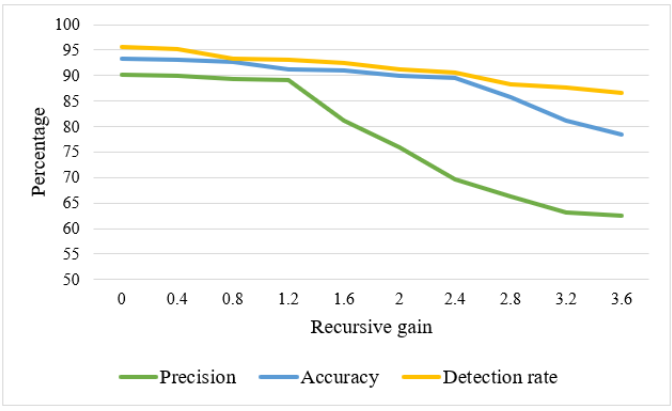

Fig. 6. Performance improvement vs. recursive gain.

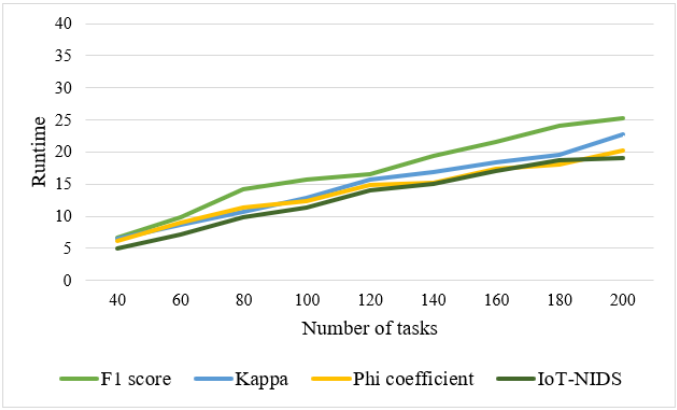

Fig. 7. Runtime vs. number of tasks.

Similarly, detected accuracy declines as the recursive gain rises. Furthermore, Fig. 7 presents a runtime comparison among models, including IoT-NIDS, F1-score, Kappa, and phi coefficient. The data indicates that as the number of tasks increases, the runtimes of these models also increase. For a task comprising 550 iterations, the runtime percentages are as follows: IoT-NIDS 91%, F1-score 92%, Kappa 94%, and phi coefficient 98%.

V. Discussion

The proposed Intrusion Detection System (IDS) leveraging neural network technologies specifically tailored for fog computing offers several advantages over previous approaches. Firstly, by utilizing neural networks, the proposed method can provide a more adaptive and sophisticated approach to intrusion detection compared to traditional methods. Neural networks excel at learning complex patterns and relationships in data, enabling them to effectively detect novel and sophisticated attack patterns that may evade conventional signature-based detection systems.

Secondly, the proposed IDS framework is specifically designed for fog computing environments, which present unique security challenges due to the distributed and dynamic nature of fog infrastructures. Unlike traditional IDSs that may struggle to scale and adapt to the complexities of fog environments, the proposed method is tailored to handle the intricacies of multi-cloud setups, ensuring robust security measures across interconnected fog nodes.

Furthermore, the research aims to tackle escalating security challenges arising from the interconnection of multi-cloud infrastructures. By developing and evaluating a novel IDS framework, the proposed method addresses the limitations of previous approaches by providing comprehensive coverage and protection against emerging threats in fog computing setups. This includes ensuring data integrity and user privacy, which are paramount in fog computing environments where sensitive data is often processed and transmitted across distributed nodes.

The assessment of computational complexity and scalability of the model, particularly in large-scale fog computing deployments, is crucial due to the inherent nature of fog computing environments where numerous devices and significant network traffic are involved. Our theoretical analysis underscores the potential benefits and challenges of increasing the number of fog nodes and handling higher network traffic. Distributed processing, a hallmark of fog computing, enables the division of tasks among multiple nodes, which enhances system robustness and performance. As the number of fog nodes increases, the workload distribution becomes more efficient, potentially leading to improved detection rates and reduced processing times per node. However, this advantage must be balanced against the increased communication overhead required for synchronization and data exchange between nodes. Effective load balancing and fault tolerance mechanisms are essential to leverage the benefits of additional nodes without succumbing to these challenges.

Furthermore, our experiments simulate various scenarios to evaluate the model's performance metrics, including detection rate, precision, accuracy, and runtime, under different configurations of fog nodes and network traffic levels. In scenarios with a limited number of fog nodes, the system might

struggle with high network traffic, leading to potential degradation in detection rate and increased false positives due to resource constraints. Conversely, with a higher number of fog nodes, the system can distribute the processing load more evenly, thereby improving overall detection performance. However, high network traffic poses a significant challenge regardless of node count. The system must process an increased volume of packets, which can strain computational resources and potentially lead to higher misclassification rates. This necessitates the implementation of efficient algorithms and possibly hardware accelerators, such as GPUs, to maintain high detection accuracy and low false positive rates.

In addition to theoretical analysis and simulations, our future work will involve real-world deployment and testing of the model in an actual fog computing setup. This will validate our simulation results and provide insights into real-world performance and challenges. Optimizing the model to handle high network traffic effectively will be a critical focus, ensuring it remains robust and efficient under varying conditions. Implementing advanced optimization strategies, such as dynamic load balancing, adaptive traffic management, and real-time processing enhancements, will further bolster the model's scalability and reliability. Moreover, incorporating additional performance metrics like latency, jitter, and energy consumption will provide a comprehensive assessment, enabling a deeper understanding of the model's operational efficiency and impact on overall system performance. Through these detailed assessments, the model can be fine-tuned to meet the demands of large-scale fog computing deployments, ensuring it remains a viable solution for intrusion detection in complex, distributed environments. Moreover, evaluating the model's performance on additional datasets beyond the NSLKDD, such as the KDD Cup 99 and more recent IoT-specific datasets, is imperative to strengthen the generalizability and robustness of the research findings. The KDD Cup 99 dataset, being a widely used benchmark for network intrusion detection, offers a broader range of attack types and network conditions, providing a more comprehensive evaluation platform for the model. For future work, we will conduct evaluations of the model's performance on additional datasets, such as the KDD Cup 99 and recent IoT-specific datasets. This will help to further validate the model's generalizability and robustness. By testing on these diverse datasets, we aim to assess the model's effectiveness in detecting a wider range of intrusion types and its adaptability to different network environments.

Integrating the proposed intrusion detection system (IDS) with other security mechanisms common in fog computing, such as encryption, access control, and secure communication protocols, is essential for creating a comprehensive and robust security framework. In fog computing environments, data often travels across various nodes and layers, making it susceptible to interception and unauthorized access. By incorporating encryption, data integrity and confidentiality can be maintained, ensuring that even if data packets are intercepted, they cannot be easily deciphered by malicious entities.

Access control mechanisms further enhance security by ensuring that only authorized users and devices can access specific resources and data within the fog network. This limits the potential attack surface and reduces the risk of unauthorized access, thereby complementing the IDS by providing an additional layer of defense. Secure communication protocols are crucial for safeguarding data as it moves between fog nodes and from edge devices to the cloud. These protocols prevent man-in-the-middle attacks and ensure that data remains secure during transit.

By integrating the IDS into a holistic security framework that includes these mechanisms, the overall security posture of the fog computing environment is significantly strengthened. The IDS can provide real-time monitoring and detection of intrusion attempts, while encryption, access control, and secure communication protocols work together to protect against data breaches and unauthorized access. This multi-layered approach ensures comprehensive protection, addressing various security challenges inherent in fog computing. Such integration not only enhances the effectiveness of the IDS but also demonstrates its practical applicability in real-world scenarios, making the research more impactful and relevant. This extended approach would be a valuable addition to the paper, showcasing a thorough and practical security solution for fog environments.

VI. Conclusion

This paper introduced a sophisticated, multi-layered neural network model designed to bolster fog computing security, particularly concerning end-users and IoT devices. It proposes an intrusion detection model aligned with fog networking to enhance the security of IoT networks. The model suggests a discontinuous neural structure refined using a modified backpropagation algorithm. The evaluation of its efficiency highlights the superiority of this adaptable structure, employing a recursive neural network, where each network is dynamically adjusted across various parameters to enhance intrusion detection. The proposed IDS model presented in this study can identify high-sensitivity task assaults, particularly those disrupting the IoT network, apart from recognizing various classes of attacks. Consequently, the model is designed to operate effectively and efficiently under continuous operational scenarios.

References

[1] V. Hayyolalam, B. Pourghebleh, A. A. P. Kazem, and A. Ghaffari, "Exploring the state-of-the-art service composition approaches in cloud manufacturing systems to enhance upcoming techniques," The International Journal of Advanced Manufacturing Technology, vol. 105, no. 1-4, pp. 471-498, 2019.

[2] B. Pourghebleh, A. A. Anvigh, A. R. Ramtin, and B. Mohammadi, "The importance of nature-inspired meta-heuristic algorithms for solving virtual machine consolidation problem in cloud environments," Cluster Computing, pp. 1-24, 2021.

[3] K. B. Raju, S. Dara, A. Vidyarthi, V. M. Gupta, and B. Khan, "Smart heart disease prediction system with IoT and fog computing sectors enabled by cascaded deep learning model," Computational Intelligence and Neuroscience, vol. 2022, 2022.

[4] V. Hayyolalam, B. Pourghebleh, M. R. Chehrehzad, and A. A. Pourhaji Kazem, "Single-objective service composition methods in cloud manufacturing systems: Recent techniques, classification, and future trends," Concurrency and Computation: Practice and Experience, vol. 34, no. 5, p. e6698, 2022.

[5] M. Abd Elaziz, M. A. Al-qaness, A. Dahou, R. A. Ibrahim, and A. A. Abd El-Latif, "Intrusion detection approach for cloud and IoT environments using deep learning and Capuchin Search Algorithm," Advances in Engineering Software, p. 103402, 2023.

[6] M. Mohseni, F. Amirghafouri, and B. Pourghebleh, "CEDAR: A cluster-based energy-aware data aggregation routing protocol in the internet of things using capuchin search algorithm and fuzzy logic," Peer-to-Peer Networking and Applications, pp. 1-21, 2022.

[7] T. Geetha and A. Deepa, "A FKPCA-GWO WDBiLSTM classifier for intrusion detection system in cloud environments," Knowledge-Based Systems, vol. 253, p. 109557, 2022.

[8] H. Lin, Q. Xue, J. Feng, and D. Bai, "Internet of things intrusion detection model and algorithm based on cloud computing and multi-feature extraction extreme learning machine," Digital Communications and Networks, vol. 9, no. 1, pp. 111-124, 2023.

[9] A.-R. Al-Ghuwairi, Y. Sharrab, D. Al-Fraihat, M. AlElaimat, A. Alsarhan, and A. Algarni, "Intrusion detection in cloud computing based on time series anomalies utilizing machine learning," Journal of Cloud Computing, vol. 12, no. 1, p. 127, 2023.

[10] 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 Computing, vol. 26, no. 1, pp. 367-384, 2023.

[11] Y. Zhong and S. Zhong, "Track Signal Intrusion Detection Method Based on Deep Learning in Cloud-Edge Collaborative Computing Environment," Journal of Circuits, Systems and Computers, p. 2350267, 2023.

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

[13] Abusitta, Adel, Glaucio HS de Carvalho, Omar Abdel Wahab, Talal Halabi, Benjamin CM Fung, and Saja Al Mamoori. "Deep learning-enabled anomaly detection for IoT systems." Internet of Things 21 (2023): 100656.

[14] Gupta, Lav, Tara Salman, Ali Ghubaish, Devrim Unal, Abdulla Khalid Al-Ali, and Raj Jain. "Cybersecurity of multi-cloud healthcare systems: A hierarchical deep learning approach." Applied Soft Computing 118 (2022): 108439.

[15] Telikani, Akbar, Jun Shen, Jie Yang, and Peng Wang. "Industrial IoT intrusion detection via evolutionary cost-sensitive learning and fog computing." IEEE Internet of Things Journal 9, no. 22 (2022): 23260-23271.

[16] Sahar, Nausheen, Ratnesh Mishra, and Sidra Kalam. "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: ICBMA 2019, pp. 39-50. Springer Singapore, 2021.