International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

A Deep Learning and Optimization-Aided Intrusion Detection Framework for Adaptive Threat Detection in Dynamic Public Cloud Environments

**[1]Dr. Banitamani Mallik, [2]Dr. Kilaru Madhavi, [3]Dr. K. Anuradha, [4]Dr. D. Kavitha, [5]Mrs. K. Prathibha, [6]Mr. Nanda Kumar Enjeti

Submitted: 03/02/2024 **Revised**: 11/03/2024 **Accepted**: 17/03/2024

Abstract: The article proposes a novel way to combine cutting-edge Deep Learning (DL) algorithms and optimization strategies to improve the effectiveness of Intrusion Detection Systems (IDS) in cloud environments. The proposed approach aims to address common issues including overfitting, imbalanced data, and the identification of unknown attack types by combining Deep Belief Networks (DBN) with Cat Swarm Optimization assisted Long Short-Term Memory (CSO-LSTM). Utilizing DBN for feature extraction lowers dimensionality, allowing for a more thorough examination of network data, and LSTM takes care of temporal factors that are essential for identifying multi-stage assaults. System performance is improved with LSTM weight optimization using Cat Swarm Optimization. The DBN architecture and the energy dependence model provide a strong basis for identifying latent patterns in the NSL-KDD dataset. This research tries to improve the adaptability and accuracy of intrusion detection systems (IDS) by integrating deep learning (DL) and optimization approaches in a comprehensive manner, in addition to addressing the current limits in IDS for cloud environments. The findings show encouraging gains in precision and flexibility, indicating the possibility for the suggested framework to overcome current intrusion detection for cloud environments constraints.

Keywords: *Intrusion Detection Systems, Cloud Computing, Deep Learning, Deep Belief Networks, Cat Swarm Optimization, Long Short-Term Memory, Network Security, Cyber Threats*

Introduction

In today's digital world, cloud computing has emerged as a prominent solution for accessing applications, data, storage, and development tools via the internet. This technology enables businesses and individuals to leverage the services provided by Cloud Service Providers (CSPs). Cloud servers offer numerous advantages, including reduced Information Technology (IT) costs, enhanced agility, time efficiency, and user- friendliness. Virtualization plays a crucial role in maximizing the utilization of data center resources by CSPs [1]. Consequently, many organizations have employed cloud computing for their infrastructures. However, the cloud environment also presents certain challenges, including data security, compliance, data mobility, vulnerability to cyber-attacks, and connectivity issues.

To enhance the security of cloud environments and mitigate the risk of DDoS attacks and other common cyber threats, organizations commonly utilize network firewalls, Intrusion Detection Systems (IDSs), and Intrusion Prevention Systems (IPSs). In the cloud environment, both IDS and IPS play crucial roles in identifying and mitigating malicious activities, including DDoS attacks. IDS functions as a network security system by continuously monitoring network traffic to detect suspicious behavior that may indicate a DDoS attack (Patel et al. 2013) [2]. By configuring the IDS to identify specific types of traffic, such as large packets, high traffic volumes or malicious traffic entering the cloud, potential threats can be promptly identified. IPS takes proactive measures to block malicious traffic before it can reach the cloud, thus preventing potential damage. Together, IDS and IPS provide a robust defense mechanism against cyber threats in the cloud environment.

Intrusion detection and prevention systems in cloud environments are increasingly utilizing Deep Learning (DL) techniques. DL algorithms, such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN)

**[1]*Professor of Mathematics, School of Applied Sciences, Centurion University of Technology and Management, Gajapati, Odisha, India, Pincode: 761211. Email: banita.mallik@cutm.ac.in*
[2]Assistant Professor, Department of Humanities and Sciences, VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad, Telangana, India, Pincode: 500090. Email: madhavi.aditi@gmail.com
[3]Research Scholar, Department of CSE, Centurion University of Technology and Management, Gajapati, Odisha, India, Pincode: 761211. Email: anukeshav76@gmail.com
[4]Associate Professor,Department of Mathematics, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences (SIMATS), Chennai , Tamilnadu, India, Pincode: 602105, Email id: soundarkavitha@gmail.com
[5]Assistant Professor, Department of EEE, Sri Venkateswara College of Engineering, Tirupati, Andhra Pradesh, In India, Pincode: 517507, Email: prathibha1122@gmail.com
[6]Assistant Professor, Department of EEE, Sri Venkateswara College of Engineering, Tirupati, Andhra Pradesh, India, Pincode: 517507. Email: e.nanda.k@gmail.com

are employed to analyze network flow data and identify anomalous patterns that may indicate malicious activities. By leveraging DL, these techniques can recognize unexpected data access requests, unexpected changes in system configuration files, malicious traffic patterns, and malicious payloads. This enables IDS and IPS to effectively detect and respond to DDoS attacks in cloud environments. In this context, the development of an efficient IDS and IPS provides essential security measures to safeguard cloud infrastructure and data.

Cloud computing is a highly convenient and accessible platform that offers on-demand access to shared computing resources, such as service applications, networks, storage, and servers. This platform enables quick provisioning and release of resources with minimal involvement from service providers, resulting in low management efforts. Within the cloud computing environment, various services are available to users (Hashizume et al. 2013) [3]. One such service is Infrastructure as a Service (IaaS), which grants users complete control over Virtual Machine (VM). Popular IaaS solutions include Open Nebula and Eucalyptus. Platform as a Service (PaaS) is another approach, allowing users to deploy their applications within the cloud platform with support from providers for languages such as Application Programming Interface (API). PaaS also encompasses application creation tools, such as Microsoft's Azure and Google App Engine. Additionally, Software as a Service (SaaS) is a service model that enables users to execute applications provided by service providers, like Google applications, via the Internet.

A Cloud Intrusion Detection System (CIDS) is a security mechanism designed to detect and respond to malicious activities or unauthorized access attempts in a cloud computing environment. ClDS systems play a crucial role in maintaining security, detecting attacks, ensuring the integrity of cloud-based services, and protecting sensitive data stored and processed within the cloud infrastructure. Chiba et al. (2016) examined the positioning, detection time, and data sources of IDS within the cloud infrastructure [4]. Through the evaluation of these factors, the evaluation aimed to identify intrusions and assess the extent to which these systems fulfill the security requirements of cloud computing environments. This analysis provided valuable insights into the effectiveness of IDS in detecting and mitigating security threats in cloud environments.

In the field of cloud computing platforms (Wen et al. 2022) presented a research study on intrusion detection technology [5], specifically focusing on the utilization of Back Propagation (BP) and the integration of Neural Network (NN) based cloud computing intrusion detection technology, and the utilization of the Artificial Bee Colony Optimization (ABCO) algorithm are employed to conduct experiments on intrusion detection to make the system effective. The IDS is positioned in the network where the cloud is connected to the internet, enabling it to provide services to cloud users are shown in Figure 1. The IDS scans the network flow data and detects whether the data packet originates from potential intruders attempting an attack or if it is a normal data packet. If the packet is determined to come from intruders with malicious intent, an alert message is generated.

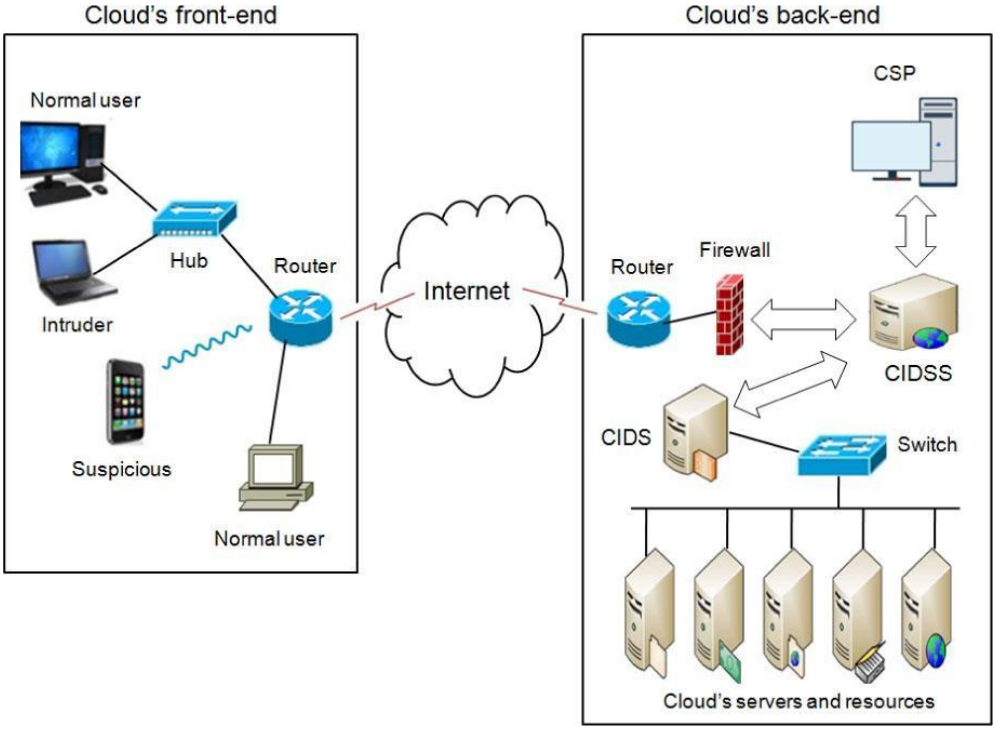

Fig 1. Intrusion Detection in Cloud Scenario

Deep learning approaches for CIDS refer to the utilization of DL techniques and algorithms to enhance the detection of intrusions in cloud computing environments. Recently, a novel DL model for cloud security, specifically aimed at cloud intrusion detection. The model combines the DL techniques of CNNs and RNNs to effectively detect and prevent unauthorized traffic within the cloud environment. The researchers conducted training and testing of their proposed model using the NSL-KDD (Network Security Laboratory - Knowledge Discovery in Databases) train dataset, a widely used benchmark dataset for cloud intrusion detection research (Hizal et al. 2021) [6].

A new intrusion detection scheme specifically designed for the cloud environment (Kumar et al. 2022) [7]. The scheme utilizes a DL technique called fuzzy min-max NN-based IDS. The methodology employed the fuzzy min-max learning algorithm, which enables the learning of nonlinear class boundaries in a single pass through the data. This algorithm also provides the flexibility to incorporate new classes and improve existing ones without the need for retraining.

Deep Neural Networks (DNNs) have emerged as a powerful approach to cloud intrusion detection. These networks leverage the capabilities of DL to automatically learn and extract complex patterns and features from network data. DNNs consist of multiple layers of interconnected neurons that process input data and make predictions. In the context of CIDS, DNNs can effectively analyze network traffic and identify potential intrusions or anomalous behavior.

A combination of a DNN with an enhanced Genetic Algorithm (GA) is proposed in a hybrid framework (Chiba et al. 2019) [8]. This hybrid approach utilizes parallel processing and incorporates Fitness Value Hashing (FVH). The framework finds practical applications in various fields such as intrusion detection, speech recognition, and computer vision. The SI algorithm, inspired by the collective behavior of organisms like bees, termites, and wasps, mimics their swarm behavior in foraging resources, nest construction, and environmental adaptation. This algorithm focuses on diverse responses, adaptability, stability, proximity, and quality to address detection issues. The self-organized emergence of a solution is achieved through the autonomous behavior of each organism within the swarm. In the context of cloud computing, virtualization is a key concept that enables cost-effective operations and efficient resource utilization. Computational intelligence-based algorithms can play a significant role in designing IDS and virtual machine placement within the virtualization process.

With the continuous growth of Internet traffic and the emergence of new threats in the cloud environment, intrusion detection has become increasingly challenging. Geetha & Deepa (2022) proposed an innovative approach for intrusion detection, combining a fisher kernel using Principle Component Analysis (PCA) for a dimensionality reduction algorithm and a GWO based weight dropped Bi-Long Short Term Memory (Bi-LSTM) classifier [9]. In their method, the PCA algorithm is applied to the data records, and the fisher kernel, incorporating the fisher score, is utilized to achieve linearly separable dimensionality reduction. This step helps in reducing the complexity of the data while preserving the relevant information. Next, the combined approach of Bi-LSTM is employed to capture long-term dependencies and extract features in both backward and forward directions. This bidirectional approach enhances the network's ability to understand sequential patterns and dependencies within the data. To optimize the recurrent weights of the Bi-LSTM network, the GWO is utilized. The GWO algorithm fine-tunes the network parameters to improve the accuracy of the classification results [10]. The classifier is trained to accurately classify the input data into either normal or different types of attacks, enabling effective intrusion detection.

Research Objective

The main objective of the research work is to develop an intelligent intrusion detection and prevention system for the cloud environment. This system utilizes Deep Belief Network (DBN)-aided Cat Swarm Optimization with Long Short-Term Memory (CSO-LSTM) detection model. Additionally, a prevention system is incorporated, which utilizes a rule engine-based prevention technique. The integration of this model and technique aims to enhance the overall effectiveness of intrusion detection and prevention, contributing to the security of the system.

Related Works

Cloud computing has become a widely adopted platform for data sharing due to its scalability and user-friendly features. However, this platform is susceptible to various intrusion attacks, including the creation of backdoors and the installation of malicious software. This raises concerns about data protection and necessitates the use of IDS as a defensive layer. The distributed nature of the cloud environment makes it attractive and vulnerable to attackers. IDS and IPS play a crucial role in enhancing system security by systematically analyzing network traffic, system logs, and configurations.

Yu et al. (2013) conducted a study to analyze files obtained from cloud servers. They applied various machine learning techniques to the dataset, including Naive Bayes (NB), Random Forest (RF), and SVM algorithms [11]. The aim of leveraging these machine learning algorithms was to enhance the detection accuracy of DDoS attacks in cloud servers. Their research objective focused on identifying DDoS attacks, and to accomplish this, they employed an open-source rule-based engine tool. Aldribi et al. Jaber &

Rehman (2020) presented an IDS designed to enhance accuracy in cloud computing environments [12]. By combining the fuzzy c-means clustering algorithm with SVM techniques, they propose a novel approach. Through implementation and evaluation using the NSL-KDD dataset, the system is compared to existing methods and the system provides a better result.

In the study, Wang et al. (2018) presented a centralized HIDS framework designed to improve resource efficiency [13]. The framework employs the Logstash tool to collect system logs from each VM, which are then stored centrally in an elastic search cluster. The logs are subsequently analyzed in a detention center, and the results are communicated back to the respective VMs. The proposed framework has been successfully validated on the OpenStack platform, with the results indicating strong performance. Aljurayban & EmamSaleh (2015) introduced a novel and efficient framework called the Layered Intrusion Detection Framework (LIDF) for identifying normal traffic within cloud computing [14]. The framework is designed to be applied across various layers of the cloud computing architecture. It leverages data mining techniques, particularly Artificial Neural Networks (ANNs), to achieve high accuracy, speed, and scalability. By incorporating ANNs, the LIDF effectively reduces the rate of analyzed traffic while enhancing throughput and overall performance. Through the utilization of the LIDF, cloud environments can better identify normal traffic patterns among monitored cloud traffic.

The literature survey of earlier works on CIDS has identified limitations, such as difficulty in identifying various types of attacks that target the network, hosts, and the application layer of the cloud. The proposed work aims at enhancing the effectiveness of CIDS by identifying types of attacks in cloud environments.

Cloud intrusion detection using DL techniques involves leveraging advanced machine learning algorithms, particularly deep learning models such as DNNs, CNNs, RNNs, and Auto-encoders. These models are widely employed to detect and address intrusions within cloud computing environments. Bhardwaj et al. (2020) developed an innovative architecture for detecting DDoS attacks in cloud environments [15]. Their approach combines an auto-encoder for feature learning with a DNN for classification, with a specific focus on differentiating between benign traffic and DDoS attack traffic. The researchers addressed various challenges related to DDoS detection by optimizing the parameters of the auto-encoder and DNN using well-designed techniques. These optimizations aimed to achieve low reconstruction error, prevent issues like exploding and vanishing gradients, and create a smaller network to mitigate overfitting. The effectiveness of the proposed approach was demonstrated through experiments conducted on the CICIDS2017 and NSL-KDD datasets.

Muthukumar & Rajendran (2015) employed an intelligent technique for building IDS with enhanced security in cloud computing environments [16]. Their approach focused on addressing both security concerns and performance issues. By incorporating intelligent algorithms into the IDS, the authors sought to improve its ability to detect and prevent malicious activities.

Latanicki et al. (2010) introduced an intelligent security model that addresses the detection of DoS attacks [17]. This model combines feature selection techniques and classification methods to effectively identify and mitigate such attacks. In the specific context of application-layer attacks, where malicious requests target vulnerabilities in the application itself, the prompt implementation of rapid response systems becomes crucial. Application-layer attacks often aim to overwhelm the targeted with a large number of requests, exhausting its resources and causing service disruption.

Rajendran et al. (2019) proposed a dedicated detection scheme for multi-stage attacks in cloud computing environments [18]. This scheme utilizes a multi-layered LSTM network, which is a type of RNN capable of capturing historical data while retaining current information. It effectively detects multi- stage attacks by analyzing the sequence of events and patterns in the cloud environment. By considering the temporal aspect of the data, the LSTM network can identify suspicious activities that may span multiple stages of an attack.

The literature survey on CIDS based on DL approaches reveals certain limitations, such as over-fitting imbalanced data and lower accuracy in detecting unknown attack types. In this proposed study, these limitations are addressed by incorporating DL with IDS techniques in cloud environments and aim at improving the training time of the model and balancing the dataset to enhance the detection rate of malicious packets.

Materials and Methods

An efficient IDS constitutes a significant contribution of the proposed framework aimed at identifying malicious packets in the cloud environment. The framework combines DL and optimization techniques to enhance the detection capabilities of IDSs, resulting in improved accuracy, faster detection, and reduced false positive rates. This integration provides a more efficient and effective security solution tailored for cloud environments.

In this research work, a DBN-based CSO-LSTM model is proposed to identify and classify malicious packets in a cloud environment. The primary objective of this work is to dimensionality reduction and it captures the most salient features of the data while discarding irrelevant or redundant information using DBN for feature extraction. The CSO-aided LSTM detection model effectively extracts the

relevant features and optimizes weights in the LSTM network to find the global optimal solution. The CSO-LSTM network aims to minimize the error rate and classify malicious packets with a high detection rate. In this chapter, the framework for DBN based CSO-LSTM model is designed to improve the accuracy and effectiveness of the detection model.

Deep Belief Network (DBN) Aided Cat Swarm Optimization with Long Short-Term Memory (CSO-LSTM)

The DBN-based CSO-LSTM model involves pre-processing raw cloud network traffic profiles, which includes label encoding and rescaling, to ensure compatibility for subsequent analysis. The rescaled data is then fed into a feature extraction subsystem, specifically using the DBN technique, to extract relevant features. To enhances the accuracy and speed of malicious packet identification, the proposed IDS model combines the CSO-based LSTM NN. This integration reduces the risk of over-fitting, ultimately improving the effectiveness of the IDS model by detecting attacks and minimizing the error rate while achieving a high detection rate.

The DBN-based feature extraction technique is employed to reduce the dimensionality of the processed data in the NSL-KDD dataset. It utilizes DBNs, which are composed of multiple layers of RBMs Hinton et al. (2012) [19]. These RBMs are stacked and trained in a greedy manner to capture the underlying structure of the dataset and detect non-linear patterns in the data. By doing so, it enhances the system's ability to uncover hidden patterns, correlations, and meaningful relationships among the variables.

The DBN is a learning method with a deep structure, known as a deep belief network, which successfully applies profound training during the training phase. When compared to CNNs, the DBN model demonstrates better representation and modelling capacity, especially in dealing with complex approximations [20].

The DBN model can be viewed as a collection of modules enclosed and stacked, forming a Boltzmann structure [21] as depicted in Figure 2. The hidden units model the dependency among the features, resulting in characteristic dependence. The DBN-based feature extraction technique reduces dimensionality and extracts meaningful features from the NSL-KDD dataset. It improves the system's ability to detect hidden patterns and relationships, enhancing the effectiveness of intrusion detection in network environments.

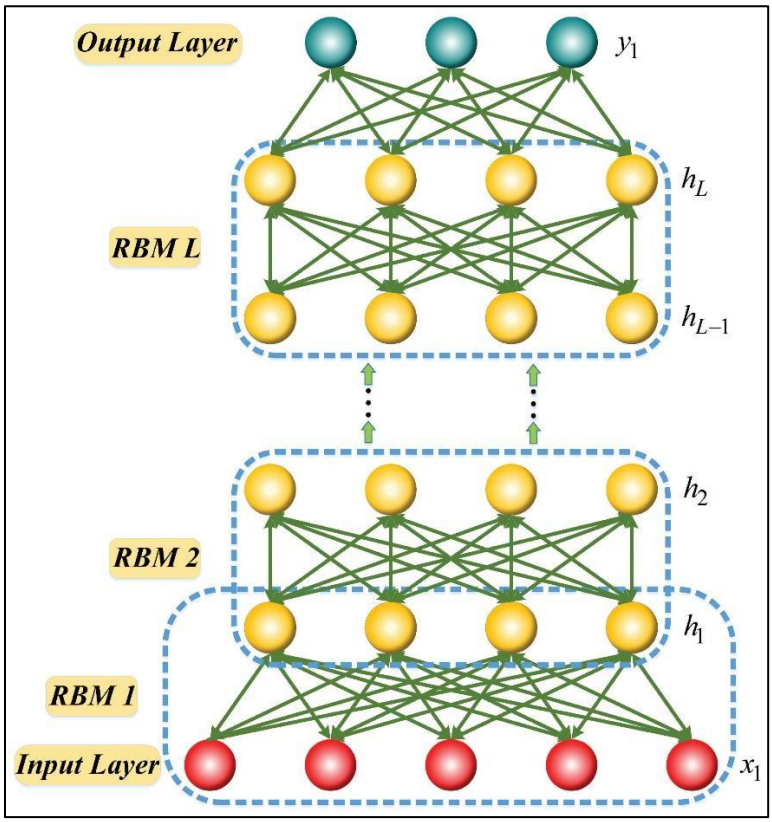

Fig 2. Architecture of DBN having RBMs for Feature Extraction

The energy dependence model ensures that all variables v and h are normally distributed through this entropy function. This function is defined by Equation (1):

$$E(m, n; \theta) = E(v, h; \theta) = -\sum_{i=1}^{m} \sum_{j=1}^{n} w_{ij} v_i h_j - \sum_{i=1}^{m} b_i v_i - \sum_{j=1}^{n} a_j h_j \quad \ldots\ldots (1)$$

where,

- E(m, n; θ): E represents the energy function where m and n represent the dimensions of the input vectors θ represents the model parameters.
- The term E(v,h;θ) represents the energy function where v and h are input variables, and θ represents the model parameters. This function quantifies the discrepancy between the predicted values and the actual values of v and h.
- In the term, $\sum_{i=1}^{m}\sum_{j=1}^{n} w_{ij}\, v_i h_j$ represents the weighted sum of the product of input vectors v and h. Here, m and n represent the dimensions of the vectors v and h, respectively. The value w_{ij} represents the weight between the ith element of v and the jth element of h. indices i and j represents the indices of the elements in the weight matrix w.
- $\sum_{i=1}^{m} b_i\, v_i$ represents index i represents the indices of the elements in the vector v. The term b_i represents the bias associated with the ith element of vector v.
- $\sum_{j=1}^{n} a_j\, h_j$ represents index j represents the indices of the elements in the vector h. The term a_j represents the bias associated with the jth element of vector h.

$$P(v, h; \theta) = \frac{e^{-E(v,h;\theta)}}{\sum_{v,h} e^{-E(v,h;\theta)}} \quad \ldots\ldots\ldots\ldots\ldots\ldots (2)$$

Where Equation (2) utilizes entropy, probabilities can be assigned to each neuron pair in both the visible and hidden layers of the network and it represents the probability distribution function for variables v and h, given parameters θ. The equation calculates the probability of observing a specific configuration of v and h by taking the exponential of the negative energy and dividing it by the sum of exponential energies over all possible configurations. This normalization ensures that the probabilities sum up to 1, making it a valid probability distribution.

The intrusion detection using the NSL-KDD dataset, a DL model is constructed effectively handles the variations in various types of attacks. RNNs are employed as a DL technique to process the intrusion data. RNN models leverage internal memory to handle input sequences of arbitrary length, enabling effective model processing. However, RNN models face issues such as the vanishing/exploding gradient problem.

To address these challenges, the LSTM model [22] is designed to overcome the long-term dependence issue in RNN models. Retrieving long-term information is crucial for effective learning, which the LSTM model aims to resolve. LSTM, as a special type of RNN, rectifies issues like the network state vanishing problem and exhibits dynamic behavior. LSTM cells replace the RNN layer cells to incorporate long-term memory functionality. The LSTM model structure includes the use of forgotten state and input gate parameters. The LSTM neural network architecture [23] is illustrated in Figure 3.

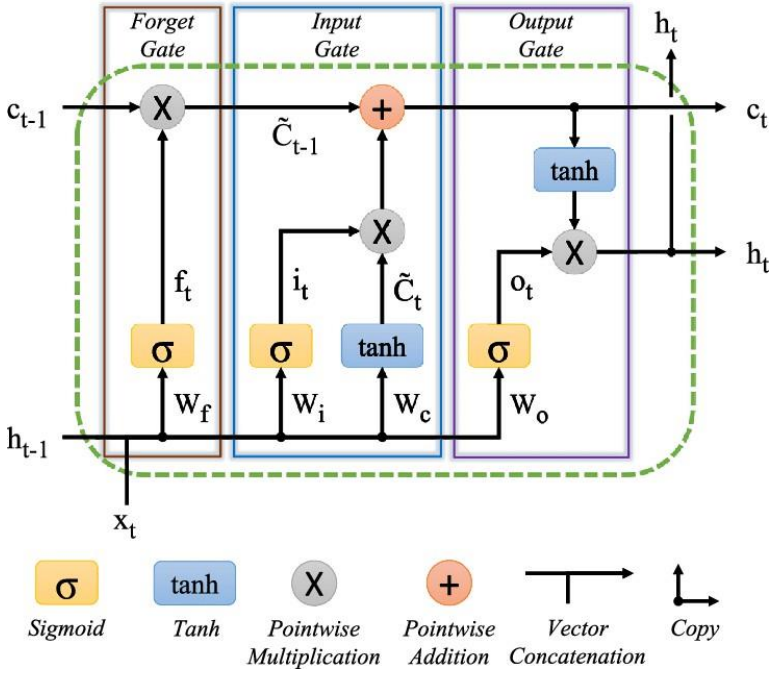

Fig 3. Architecture of LSTM Neural Network

Compared to traditional machine learning methods, LSTM networks) excel at identifying repetitive attack patterns in long sequences of data packets, regardless of the window size used. The forward calculation technique in the LSTM neural network is described by equation (5.3) below:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_c[h_{t-1}, x_t] + b_c)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

$$o_t = \sigma(\varpi_v[h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(C_t) \ldots\ldots\ldots\ldots\ldots\ldots (3)$$

where,

- bias matrix is represented by W and the bias terms are denoted by b.
- The forget gate is denoted by ƒt, the input gate by it, and the output gate by o_t, $\tilde{C}t$ by cell state, tanh by hyperbolic tangent activation function and
- σ sigmoid function and the hyperbolic tangent activation function tanh.

Equation (3) demonstrates how the LSTM model addresses the long-term dependence problem in RNN models. Consequently, this prediction phase of the NN relying on the LSTM model, can effectively identify and detect DDoS attacks in a cloud environment.

Algorithm 1: Feature Selection in IDS

Input: No. of Cats, Fitness Function, No. of Iterations, Inertia Weight.

Output: $gbest_i$, $pbest_i$

Step 1: The positions and velocities of each cat (features) are randomly initialized during Swarm initialization within the data packets

Step 2: If the fitness function value of cat x_i is greater than its personal best ($pbest_i$)

Update $pbest_i = x_i$

Step 3: If the fitness value of $pbest_i$ is greater than the global best($gbest_i$) update $gbest_i = pbest_i$.

Step 4: Calculate the new velocity v_{id}^{t+1} of the cat using the formula:

$$v_{id}^{t+1} = w * v_{id}^{t} + c_1 * r_{1i} * (P_{id}^{t} - x_{id}^{t}) + c_2 * r_{2i} * (P_{gd}^{t} - x_{id}^{t})$$

Step 5: Update the cat's position as follows:

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1}$$

Step 6: If the stopping criteria are not met, repeat steps 2-5.

Step 7: Return the gbest value and its fitness function value for the IP packet features.

The CSO optimization-based LSTM neural network detection model utilizes the CSO algorithm to enhance the accuracy of attack prediction through weight optimization in an LSTM neural network. This hybrid approach combines the strengths of CSO and LSTM to overcome issues related to network convergence and improve the overall performance of intrusion detection.

In this model, the weights of the LSTM hidden layer serve as inputs to the CSO algorithm. The output error of the original LSTM is used as a measure of Cat Swarm fitness, indicating the effectiveness of the cats [24]. By optimizing the weights through the CSO algorithm, the model aims to improve the accuracy of attack prediction. This proposed model is evaluated using the NSL-KDD dataset. It leverages the capabilities of LSTM neural networks to capture temporal patterns and detect anomalies in network traffic.

$$\mathcal{V}_i^{k+1} = \omega\mathcal{V}_i^{k} + l_1 r_1(P_i^{k} - X_i^{k}) + l_2 r_2(P_g^{k} - X_i^{k})$$

$$X_i^{k+1} = X_i^{k} + \mathcal{V}_i^{k+1} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4)$$

The process of segmenting the input data, feeding it into a hidden layer consisting of LSTM cells, and calculating the hidden states of the LSTM cells. The LSTM cells analyze and capture temporal dependencies within the segmented data, producing hidden states that represent encoded given in Equations (5), (6), and (7):

$$X = (X_1, X_2, \ldots, X_L) \ldots\ldots\ldots\ldots (5)$$

$$X_t = (f_t' + f_{t+1}', \ldots, f_{m-L+t-1}'); 1 \le t = L; p, L \in N \ldots\ldots\ldots\ldots\ldots (6)$$

$$h_t = LSTM_{f0}(X_t, c_{t-1}, h_{t-1}) \ldots\ldots\ldots\ldots\ldots\ldots (7)$$

The error calculation formula is the mean square error, resulting in the loss function defined by Equation (8).

$$LSTM_{LOSS} = \sum_{i=1}^{L(m-L)} \frac{(h_t - v_t)^2}{(L(m-L))} \ldots\ldots\ldots\ldots (8)$$

Algorithm 2: DBN-Aided CSO-LSTM Detection Model

Input: **Pre-processed Traffic Profiles**

Output: **Normal or Malicious Packets**

//Deep Belief Network for Feature Extraction

Step 1: Initialize the Deep Belief Network model with hidden layers, learning rate, batch size, and activation function.

Step 2: Extract features using the trained DBN network

//CSO based LSTM Neural Network for Attack Detection

Step 3: Define the CSO function with features, num_cats, num_iterations.

Step 4:	Evaluate the fitness of each cat for p in cats.
Step 5:	Update the cat velocities and positions.
Step 6: Determine the best cat.	
Step 7:	Update the best cat with high fitness value and return the best cat's weights.
Step 8:	Define the LSTM function with best cats weights.
Step 9:	Create the CSO-LSTM model with hidden layers and activation function.
Step 10: Compile the model with loss='binary_crossentropy', optimizer='Adam' and metrics= 'accuracy'.	
Step 11: Fit and evaluate the trained DBN and CSO-LSTM detection model.	
Step 12: Detected normal or malicious packets with accuracy, precision, recall, and f-measure	

The results obtained from an efficient IDS using DBN feature extraction and CSO-LSTM detection model Subsystem are found to be better in terms of detection rate.

Results and Discussion

In this section, it presents the results obtained from DBN feature extraction and a CSO-LSTM detection model. The experiments were conducted using the NSL-KDD dataset, and pre-processing techniques such as label encoding and rescaling were applied to develop the system. In the context of intrusion detection system evaluation metrics, the following performance metrics are used:

- True Negative (TN) corresponds to instances correctly recognized as non-attacks, indicating the accurate classification of benign or normal activities.
- False Negative (FN) accounts for instances mistakenly labelled as non-attacks, revealing malicious attacks that were inaccurately identified as benign or normal.
- False Positive (FP) represents instances inaccurately categorized as attacks, indicating normal or benign occurrences mistakenly flagged as malicious attacks.
- True Positive (TP) reflects instances accurately identified as attacks, demonstrating the precise classification of malicious attacks.
- Precision is the ratio of truly categorized attack samples (TP) to the sum of predictive positive samples (FP + TP), gauging the accuracy of identifying attacks while excluding false positives.
- Recall (Sensitivity) is the ratio of true positive samples (TP) to the sum of False Negatives (FN) and true positive samples (TP), assessing the system's ability to identify actual positive samples, including avoiding false negatives.
- F-Measure is the harmonic mean of precision and recall, offering a balanced assessment of both precision and recall and considering the trade-off between them.
- Accuracy measures the overall classification performance of the model, representing the proportion of correctly categorized samples (both true positives and true negatives) to the total number of samples.

A comparative study in Table 1 shows the differences between the proposed DBN-aided CSO-LSTM with other previous models and the graphical representation is depicted in Figure 4.

Table 1. Comparison of the Proposed Model with Existing Models

Performance Metrics (%)	Accuracy	Precision	Recall	F1 score
LSTM-SVM	87.09	87.78	88.09	86.83
LSTM-DNN	88.62	91.12	90.46	89.23
LSTM-RNN	89.73	90.87	91.73	90.04
DBN+CSO-LSTM	**96.53**	**94.85**	**96.78**	**94.2**

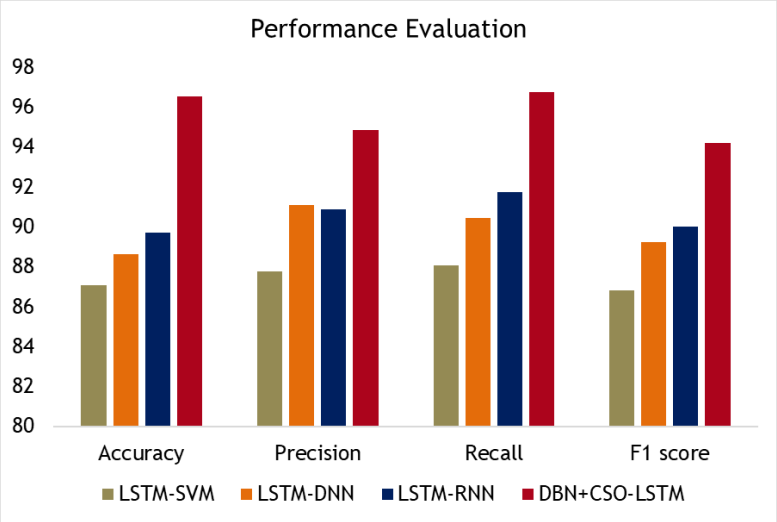

Figure 4. Performance Evaluation of Existing and Proposed Intrusion Detection Systems

The performance metrics for different intrusion detection systems are presented in terms of accuracy, precision, recall, and F1 score. In the evaluation, the LSTM-SVM model achieved an accuracy of 87.09%, with precision, recall, and F1 score values of 87.78%, 88.09%, and 86.83% respectively. The LSTM-DNN model demonstrated improved performance, achieving an accuracy of 88.62%, with high precision (91.12%), recall (90.46%), and F1 score (89.23%). Furthermore, the LSTM-RNN model exhibited even better results, with an accuracy of 89.73% and impressive precision (90.87%), recall (91.73%), and F1 score (90.04%). Remarkably, the DBN+CSO-LSTM model outperformed the other models, achieving a remarkable accuracy of 96.53%. This model also demonstrated high precision (94.85%), recall (96.78%), and F1 score (94.2%), suggesting superior performance in accurately classifying instances as attacks or non-attacks. These results highlight the effectiveness of combining deep learning techniques, such as LSTM, with innovative approaches like DBN+CSO, in enhancing the overall performance of intrusion detection systems.

Conclusion

In conclusion, a new era of technical growth has been brought about by the quick acceptance of cloud computing, which offers previously unheard-of simplicity and scalability. But the widespread use of cloud services has also resulted in serious security issues, such as the susceptibility to different cyberthreats and attacks. The vital function that IDS play in bolstering cloud environments' security posture was examined in this study. By utilizing cutting-edge methods like DL, especially with models such as CNNs, RNNs, and DBNs, IDS are better able to identify and stop unwanted activity. The models follow a trend of improving performance from LSTM-SVM to DBN+CSO-LSTM, indicating the effectiveness of the applied techniques. DBN+CSO-LSTM stands out as the top performer, achieving the highest scores across all metrics. The choice of model depends on the specific requirements and trade-offs desired, as each model excels in different aspects of intrusion detection performance.

References

[1] Tari, Z. (2014). Cloud Computing: A Primer. Chapman and Hall/CRC.

[2] Patel, M., et al. (2013). Intrusion Detection Systems: A Review. International Journal of Computer Applications, 75(13).

[3] Hashizume, K., et al. (2013). Cloud Computing Security Issues and Solutions: A Survey. International Journal of Information Management, 33(1), 13-25.

[4] Chiba, R., et al. (2016). Positioning, Detection Time, and Data Sources of Intrusion Detection Systems in Cloud Environments. Procedia Computer Science, 82, 165-172.

[5] Wen, Y., et al. (2022). Back Propagation and Neural Network-Based Cloud Computing Intrusion Detection Technology. Journal of Ambient Intelligence and Humanized Computing, 13(2), 1747-1759.

[6] Hizal, A., et al. (2021). NSL-KDD: A New Intrusion Detection Dataset for the Cloud Era. Journal of King Saud University - Computer and Information Sciences.

[7] Kumar, A., et al. (2022). Fuzzy Min-Max Neural Network-Based Intrusion Detection Scheme for Cloud Computing. Neural Computing and Applications, 1-19.

[8] Chiba, R., et al. (2019). Hybrid Framework of Deep Neural Network and Enhanced Genetic Algorithm for Intrusion Detection. IEEE Access, 7, 158336-158346.

[9] Geetha, P., & Deepa, S. (2022). Fisher Kernel-Based Weight Dropped Bi-LSTM Classifier for

Intrusion Detection. Computers, Materials & Continua, 70(1), 1065-1083.

[10] Muhuri, P. K., et al. (2020). Genetic Algorithm-Based Feature Selection for LSTM-RNN in Intrusion Detection. In Proceedings of the International Conference on Computational Intelligence and Data Engineering (ICCIDE), 1-6.

[11] Bhardwaj, A., et al. (2020). An Autoencoder-Based Deep Learning Architecture for DDoS Attack Detection in Cloud Computing. Soft Computing, 24(1), 331-341.

[12] Muthukumar, S., & Rajendran, P. (2015). An Intelligent Technique for Building Intrusion Detection System with Enhanced Security. Procedia Computer Science, 46, 866-874.

[13] Latanicki, J., et al. (2010). Intelligent Security Model for Application Layer DoS Attack Detection in Cloud Computing. In Proceedings of the International Conference on Computer Science and Information Technology (ICCSIT), 18-22.

[14] Rajendran, P., et al. (2019). Multi-layered LSTM Network for Detection of Multi-stage Attacks in Cloud Computing. Future Generation Computer Systems, 91, 442-452.

[15] Yu, L., et al. (2013). Cloud-Based Intrusion Detection System. In Proceedings of the International Conference on Cloud Computing and Big Data (CloudCom-Asia), 1-6.

[16] Aldribi, A., et al. (2020). A Novel Intrusion Detection System for Cloud Computing Environments Based on Fuzzy C-Means and SVM. Journal of Ambient Intelligence and Humanized Computing, 11(6), 2451-2464.

[17] Wang, W., et al. (2018). A Centralized HIDS Framework for Resource Efficiency in Cloud Computing. Future Generation Computer Systems, 79, 518-529.

[18] Aljurayban, S., & EmamSaleh, I. (2015). Layered Intrusion Detection Framework (LIDF) for Cloud Computing. Journal of King Saud University - Computer and Information Sciences.

[19] Fischer, A., et al. (2014). Deep Learning with Boltzmann Machines: A Learning Algorithm for the Architecture of the Human Brain. Neural Computation, 26(1), 1- 48.

[20] Hinton, G. E., et al. (2012). A Practical Guide to Training Restricted Boltzmann Machines. In Neural Networks: Tricks of the Trade (pp. 599-619). Springer.

[21] Pankajavalli, P. B., & Karthick, G. S. (2022). Efficient Data Flow Graph Modeling Using Free Poisson Law for Fault-Tolerant Routing in Internet of Things. In Computer Networks and Inventive Communication Technologies: Proceedings of Fifth ICCNCT 2022 (pp. 475-487). Singapore: Springer Nature Singapore.

[22] Al-Emadi, I., et al. (2020). A Comprehensive Review on Long Short-Term Memory Networks: From Data Preprocessing to Model Optimization. IEEE Access, 8, 144516-144543.

[23] Yu, S., et al. (2019). A Novel Intrusion Detection System for Cloud Computing Based on LSTM Networks. IEEE Access, 7, 133379-133389.

[24] Karthick, G. S. "Energy-Aware Reliable Medium Access Control Protocol for Energy-Efficient and Reliable Data Communication in Wireless Sensor Networks." *SN Computer Science* 4.5 (2023): 449.