International Journal of Management, IT & Engineering
Vol. 15 Issue 01, January 2025,
ISSN: 2249-0558 Impact Factor: 7.119
Journal Homepage: http://www.ijmra.us, Email: editorijmie@gmail.com
Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gate as well as in Cabell's Directories of Publishing Opportunities, U.S.A

DMHA-ABL: A DEEP LEARNING FRAMEWORK FOR INTRUSION DETECTION IN CLOUD INFRASTRUCTURES

Srinivas Potluri
IT Director EGS Global, California, USA.

Abstract

As cloud computing is expanding at a fast pace, securing dynamic, distributed cloud infrastructures against advanced cyberattacks has become a priority. Traditional Intrusion Detection Systems (IDS) are not flexible and less accurate in detecting evolving attack types of cloud infrastructures. This paper proposes a deep learning-based IDS framework designed for cloud platforms with an emphasis on intelligent, scalable threat detection. The suggested system employs exhaustive preprocessing techniques like data cleansing, removal of outliers using Z-score, and Min-Max normalization for enhancing the consistency and quality of data. Then, the system applies a hybrid deep learning architecture, which Densely connected Multi-Head Attention-based AlexNet combined with Bi-LSTM (DMHA-ABL), which efficiently captures intricate spatial and temporal attack patterns. For improved performance, we incorporate an Adaptive Whale Optimization Algorithm (AWOA) for dynamic hyperparameter tuning of important parameters like learning rate, batch size, and epochs. This optimization leverages a rank-based inertia weight mechanism to maintain exploration-exploitation balance, which considerably enhances detection accuracy. The framework is tested on cloud-relevant datasets to detect threats such as DDoS, illegal access, and malware spread. Experimental outcomes show that our proposed framework achieves an accuracy of 98.76%, significantly outperforming existing models

Keywords: *Cloud Computing, Anomaly detection, AlexNet, Whale Optimization Algorithm, AlexNet, Bi-LSTM.*

1 Introduction

The rapid growth of cloud computing has transformed the management of data and services, but also created new security threats [1]. Cloud environments become more vulnerable to advanced, and complex cyberattacks as they get distributed and dynamic [2]. Because traditional intrusion detection systems (IDS) are usually written for static on-premises environments, they will be behind. Such systems may lack the scalability, intelligence, and flexibility needed to identify and respond to emerging patterns of attacks

[3]. Consequently, there exists an increasing demand for constantly evolving and more adaptive security solutions expressed tailored to cloud system environments. Complex information and services are being secured against the cyberattacks of nowadays by using machine learning, deep learning, behaviour analytics, and inspection to improve detection capabilities [4].

Intrusion detection systems (IDS) are often used in the cloud environment to deploy scans of network traffic, detecting attempts at unauthorized access, and get alerts to anomalies characteristic of cyberattacks like DDoS [5]. To increase automated response functions and threat visibility. IDS method integrates with orchestration systems, security information, and event management tools with adaptive cloud deployments [6]. Machine learning IDSs are flexible and appropriate for distributed and elastic cloud environments by adapting to changing threats. Intrusion detection systems (IDS) are challenged in cloud environments with scaling issues, too many false positives, and poor visibility in multi-tenant and encrypted settings [7]. Additionally, highly dynamic cloud applications, data flows, and applications challenge static signature-based IDS to reduce their ability to detect newly evolving or hidden attack methods [8].

Furthermore, several techniques overcome the limitation, like a new machine learning technique for DDoS cloud identification: the Bayesian-based Convolutional Neural Network (BaysCNN) model. Dimensionality reduction by Principal Component Analysis (PCA) is applied to the CIDDoS2019 dataset. Improves IDS's scalability and reliability in cloud computing environments [9] and a new IDS that identifies FDIAs in smart grids through the integration of deep learning classifiers and hybrid feature selection. To better capture the smart grid, the IDS employs a hybrid deep learning classifier that integrates Long Short-Term Memory (LSTM) networks and Convolutional Neural Networks (CNN). The temporal and spatial natures of the data [10]. For the better detection of DDoS attack accuracy and flexibility, HybridNet integrates the optimal characteristics of both deep learning and machine learning methods [12].

Several researchers have proposed using machine learning techniques to find and detect network intrusions. With current machine learning-based solutions, the most common issues are as follows: To begin with, these models show a significant false positive rate for a wide range of attacks [11]. Existing research primarily uses a single dataset to evaluate the effectiveness of machine learning models, which makes the lack of generalizability evident. The size of modern network traffic is too great for the models that are currently being examined to handle [13]. Finally, technologies that are capable of handling the growing

dimensions, speed, and dynamics of today's high-speed networks are desperately needed [14]. As opposed to traditional machine learning, the new deep learning paradigm has demonstrated advanced performance in several fields, particularly intrusion detection. It offers a better way to represent data, which makes it possible to generate better models [15]. The key contribution of the work as follows:

- A robust data preprocessing framework is developed that includes data cleansing, Z-score-based outlier removal, and Min-Max normalization to ensure high-quality, noise-free, and normalized input data suitable for cloud-specific intrusion detection.
- A novel Densely connected Multi-Head Attention-based AlexNet with BiLSTM (DMHA-ABL) is introduced, which combines spatial feature extraction, attention-based feature enhancement, and temporal sequence modelling to effectively detect various cloud-based cyberattacks with improved accuracy and generalization.
- An enhanced optimization strategy using Adaptive Whale Optimization Algorithm (AWOA) with a rank-based inertia weight mechanism is employed to fine-tune critical training parameters such as learning rate, batch size, and epochs, thereby improving model convergence and overall detection performance.

These sections are presented in the following order: Section 2 explains some of the related research and literature reviews, Section 3 presents the introduced framework, Section 4 includes a comprehensive analysis of the noted results and discussions, and the last evaluation for this study is presented in Section 5.

2 Literature Review

A few of the most recent studies about intrusion detection in cloud infrastructure were reviewed in this section.

In 2022, Bahaa et al. [16] presented a new hybrid meta-heuristic adaptive particle swarm optimization whale optimizer algorithm (APSOWOA) for hyperparameter optimization of a convolutional neural network (APSOWOA-CNN). The fitness value of the APSO–WOA optimization algorithm is determined as the cross-entropy loss function of the validation set during training of the CNN model. The algorithm surpasses the FNN algorithm when training models.

In 2024, Aljuaid et al. [17] present a deep learning-based model with the utilization of convolutional neural networks (CNNs) to identify cyberattacks in the cloud. The model incorporates the utilization of the SMOTE balance data technique, feature selection, dataset gathering, preprocessing, model training, testing, and performance assessment, which has been successful in guarding cloud networks from multiple attacks.

In 2024, Najafli et al. [18] suggested in this paper, there is a study of an adaptive IDS based on deep reinforcement learning (DRL) for managing interactions with unidentified environments and executing binary detection under fog. Assembling is performed to classify the attacks into numerous classes, wherein the CICIDS2018 dataset is adopted for both multiclass and binary classification.

In 2022, Chiba et al. [19] propose a new hybrid optimization method termed "ISAGASAA," which utilizes deep neural networks in order to establish an effective anomaly network intrusion detection system. To discover the optimum set of parameters to achieve a maximal high detection rate, accuracy, and minimal false alarm rate, ISAGASAA hybridizes the Improved Self-Adaptive Genetic Algorithm (ISAGA) and the Simulated Annealing Algorithm (SAA).

In 2024, Long et al. [20] proposed a brand-new NIDS algorithm tailored to cloud environments and relying on the Transformer model. By merging the essential concepts of network intrusion detection and the state-of-the-art attention mechanism integrated in the Transformer model, our algorithm enhances detection performance by allowing a more in-depth examination of relationships between different intrusion types and input features. Conducted an extensive comparative analysis and provided a thorough design of our methodology.

In 2025, Sorour et al. [21] suggested a decentralized intrusion detection model for IoT networks using the Joint Strategy Optimization algorithm to optimize hyperparameters. The model utilizes a Long Short-Term Memory network structure to capture complex intrusion patterns and federated learning to enable collaborative training across various networks. The suggested model is evaluated using Transformer, bi-LSTM, and GRU models and offers efficient security solutions to IoT communities.

In 2025, Veeramachaneni et al. [22] proposed an Adaptive Ensemble Learning Framework to enhance the intrusion detection system's robustness in WSNs. To maximize detection accuracy and reduce false positives, the framework integrates Long Short-Term Memory (LSTM) networks and Convolutional Neural Networks (CNNs) through a dynamic voting mechanism.

In 2024, Bukhari et al. [23] proposed a new Stacked Convolutional Neural Network and Bidirectional Long Short Term Memory (SCNN-Bi-LSTM) model for intrusion detection in Wireless Sensor Networks (WSNs). It employs Federated Learning to promote performance and privacy, where several sensor nodes can train a central global model without exposing private information.

In 2024, He et al. [24] introduced a new Dynamic Graph Transformer-based Parallel Framework (DGT-PF) for effective system anomaly detection on clouds. The framework embeds spatiotemporal features of KPIs by means of Graph Neural Network and Transformer with an anomalous attention mechanism to achieve better timeliness and accuracy. Further, the method parallelizes applying the AR-MLP model, the nonlinear neural network model, and the dynamic relationship embedding strategy.

In 2024, Gwassi et al. [25] presented that the work employs CapsNet to scan employee fingerprints and blink patterns, QKSCP to diminish communication channel exposures, DQN-A3C algorithm for detection and prevention, explainable DQN-A3C model and SILT transformer for NLP explanations, Hopping IDS/IPS employing HGN model, and a Federated Learning-based global risk assessment mechanism for enhanced cyberattack understanding. Table 1 addresses the aim, methodology, advantages, and drawbacks of the existing works.

Table 1 Review of various authors about the existing works

Author	Aim	Methodology	Advantage	Drawback
Bahaa et al. 2022 [16]	To optimize CNN hyperparameters using a hybrid meta-heuristic algorithm.	APSOWOA (Adaptive PSO + Whale Optimizer), fitness based on cross-entropy loss.	Outperforms FNN in CNN training; efficient convergence.	May have high computational cost; model-specific tuning required.
Aljuaid et al. 2024 [17]	Detect cyberattacks in cloud environments using deep learning.	CNN with SMOTE for data balancing, feature selection, and preprocessing.	High detection performance; reduces data imbalance impact.	Performance depends on feature selection; lacks focus on real-time adaptation.
Najafli et al. 2024 [18]	Develop adaptive IDS using DRL for fog computing.	Deep Reinforcement Learning, ensembling, CICIDS2018 dataset.	Adaptive to unknown environments; works for both binary and multiclass tasks.	DRL models can be slow to train; complex reward shaping.
Chiba et al. 2022 [19]	Improve anomaly detection via hybrid optimization.	ISAGASAA = ISAGA (Genetic) + SAA (Annealing), deep neural networks.	High detection rate, accuracy; low false alarm rate.	Hybridization complexity; tuning of multiple parameters.

Long et al. 2024 [20]	Build a Transformer-based NIDS for cloud environments.	Transformer model with attention to explore input relationships.	Strong capability for pattern recognition; scalable.	Transformers are resource-intensive; needs large data.
Sorour et al. 2025 [21]	Intrusion detection in IoT using decentralized model.	Joint Strategy Optimization + LSTM + Federated Learning.	Enables privacy-preserving, distributed training.	May suffer from latency and device heterogeneity.
Veeramachaneni et al. 2025 [22]	Boost robustness in WSN IDS using ensemble learning.	Adaptive Ensemble: CNN + LSTM with dynamic voting.	Reduced false positives; robust detection.	Model complexity may limit deployment on low-power nodes.
Bukhari et al. 2024 [23]	Intrusion detection in WSNs with privacy preservation.	SCNN + Bi-LSTM + Federated Learning.	Preserves privacy; handles sequence data well.	High computation on edge devices; synchronization overhead.
He et al. 2024 [24]	Anomaly detection in cloud via dynamic graph-based method.	DGT-PF: GNN + Transformer + AR-MLP + dynamic embedding.	Captures spatiotemporal anomalies effectively.	High complexity; may be hard to interpret in real-time.
Gwassi et al. 2024 [25]	Cyberattack prevention using multimodal and explainable techniques.	CapsNet, QKSCP, DQN-A3C, SILT Transformer, FL, HGN.	Multi-layered protection; explainable AI; federated assessment.	Very complex system; potential integration and latency issues.

2.1 Research Gap

Despite considerable advances in IDS and cybersecurity models, there are some research gaps, as evident from the reviewed work. While many models, such as those utilizing CNNs, LSTMs, Transformers, and hybrid metaheuristic optimizers, have improved in accuracy and detection, there is still a prevailing limitation on their universal applicability and flexibility in the wake of shifting trends of cyber-attacks, particularly within resource-constrained networks like WSNs and IoT. Also, although many models do involve explainability, most deep-learning-based IDS are black-box models, and there is difficulty with interpreting predictions and ensuring trustworthiness. Excessive benchmarking on varied sets and scenarios also lacks, hindering cross-domain applicability for solution

proposals. Hence, the need for light-weight, explainable, adaptive, and decentralized IDS models exists that can thrive in heterogeneous network settings.

3 PROPOSED METHODOLOGY

The proposed method presents a hybrid deep learning-based intrusion detection system for applications in cloud setups with emphasis put on scalable as well as accurate anomaly detection. The methodology begins with rigorous preprocessing involving data cleansing to remove inconsistencies, removal of outliers through the use of Z-score-based outlier detection to filter out anomalous points, and Min-Max normalization to rescale features on a uniform basis. Then, the pre-processed information is fed into a novel deep learning architecture known as Densely connected Multi-Head Attention-based AlexNet with BiLSTM (DMHA-ABL). The architecture employs dense connections to facilitate feature reuse and efficient gradient flow, whereas the Multi-Head Attention (MHA) module enhances feature representation by collecting diverse patterns from different subspaces. The AlexNet backbone is used to extract spatial features and BiLSTM to achieve bidirectional temporal dependencies so that the model is able to detect complex intrusion patterns. SoftMax classifier and fully connected layer are used for the final prediction. For further optimization of the training process, an Adaptive Whale Optimization Algorithm (AWOA) is used to adaptively modify hyperparameters like learning rate, batch size, and number of epochs. This optimization employs a rank-based inertia weight approach in order to achieve a balance between global exploration and exploitation and hence improve convergence and classification performance. The entire methodology is implemented in Python and evaluated on cloud-related intrusion datasets. The overall architecture of the proposed method is shown in Figure 1.

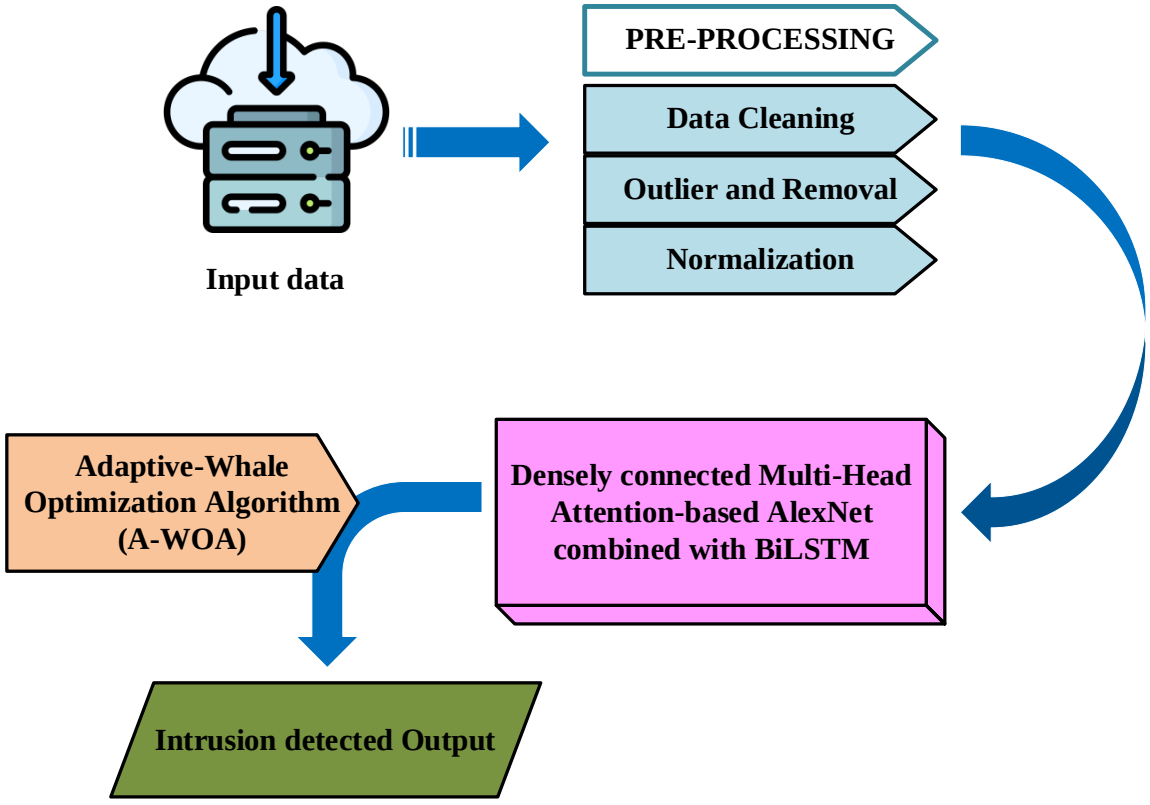


Figure 1 Architecture of the proposed methodology

3.1 Preprocessing

Preprocessing is a foundation in developing an effective intrusion detection system (IDS), especially when handling deep learning models and complex cloud infrastructure datasets. In this work, the preprocessing phase includes three core components: Data Cleaning, Outlier Detection and Removal, and Normalization. These operations ensure that the input data is consistent, noise-free, and prepared for the following feature extraction and model training procedures.

3.1.1 Data Cleaning

Data cleaning is a critical step in preparing cloud traffic data for intrusion detection. It involves removing duplicate records, correcting inconsistent or missing values, and deleting corrupted or incomplete data entries. This renders the dataset accurate, consistent, and noise-free, which would not hinder model performance. By maintaining data integrity and adherence to standard network protocols, the learning model can better distinguish between legitimate and malicious activity, leading to improved detection accuracy.

3.1.2 Outlier detection and removal

Outliers, or extreme values, may deceive learning algorithms by distorting the decision boundary, especially for sensitive detection problems such as anomaly classification. In this, apply Z-score-based outlier detection, a statistically sound approach, to detect and remove such anomalies from the training set. The Z-score for each feature is computed using the formula:

$$Z_{score}(i) = \frac{x_i - \mu}{\sigma} \quad (1)$$

Where μ = distribution mean, σ = standard deviation and x_i is the individual feature value. The standard deviation can be calculated using:

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2} \quad (2)$$

Outliers are removed by setting the threshold Z-score at ±3, meaning values greater than three standard deviations from the mean. Any point that exceeds this is discarded to build a more generalized and robust learning process.

3.1.3 Normalization

Different scales of network traffic characteristics like packet length, period, or bytes are involved; normalization is indispensable for preventing dominant characteristics from having oversized effects on the model. Under this system, we use Min-Max Normalization to rescale all the characteristics into the range [0, 1] to have uniform importance over the input space. Normalization is calculated through the equation:

$$X_{norm} = \frac{X - M_{in}(X)}{M_{ax}(X) - M_{in}(X)} \quad (3)$$

where X represents the initial value, $M_{ax}(X)$ is the highest possible value in the dataset, and $M_{in}(X)$ is the lowest amount in the dataset.

3.2 Densely connected Multi-Head Attention-based AlexNet with BiLSTM

To mitigate the limitations of the traditional intrusion detection frameworks in dynamic cloud scenarios, propose a novel deep learning-based intrusion detection framework called Densely connected Multi-Head Attention-based AlexNet with BiLSTM (DMHA-ABL). The architecture of the proposed approach has been developed to extract efficiently the spatial and temporal features associated with cloud traffic data. Leveraging the spatial feature learning capabilities of convolutional neural networks and the sequence

modeling capabilities of recurrent neural networks, the system is optimized for high-accuracy anomaly detection. The architecture for the proposed DMHA-ABL is shown in Figure 2.

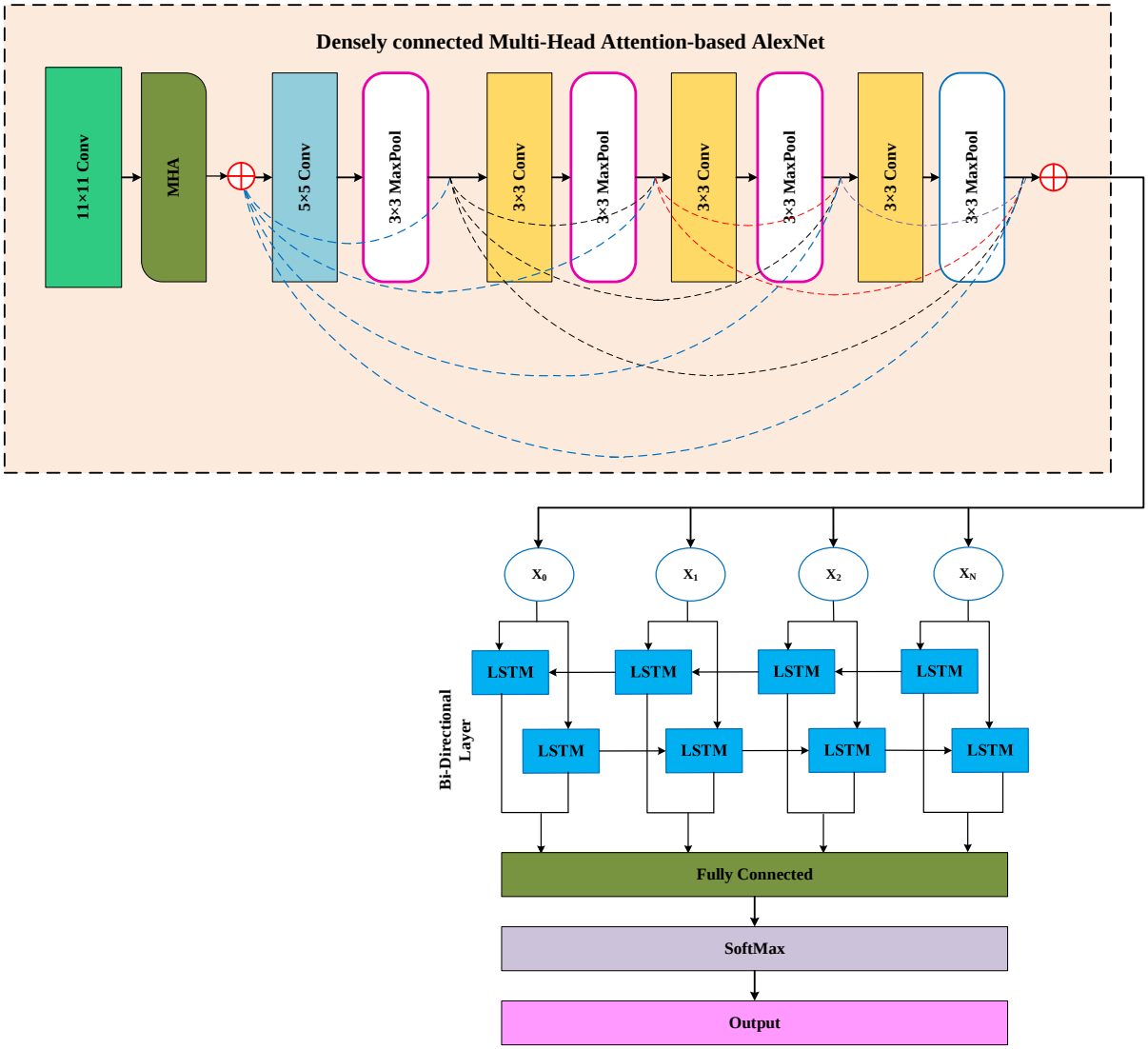


Figure 2 Architecture of DMHA-ABL

In the framework, use AlexNet as the foundational architecture due to its proven effectiveness in obtaining low- to high-level visual features. In further enhancing the attention of the model on salient spatial information, we incorporate a Multi-Head Attention (MHA) module immediately following the initial 11×11 convolution layer. The attention mechanism allows the network to pay attention to multiple feature representations simultaneously, increasing its ability to pay attention to key areas of the input and strengthening the robustness of feature learning. After MHA application, the output is concatenated to the initial convolutional features and then passed into a series of convolutional layers with 5×5 and 3×3 kernels. Every one of the convolutional blocks is preceded by a 3×3 MaxPooling layer to downscale the feature maps without significant

activations loss. Add dense connections between all the convolutional layers importantly. These connections offer each layer inputs from all the previous layers, which encourages reuse of features, Aids in mitigating vanishing gradients, and Enables deeper supervision, thus increasing information and gradient flow inside the network. Dense connectivity plays a significant role in enhancing learning efficiency and feature propagation across the network. To find temporal patterns and long-range dependencies in network traffic patterns, the spatially processed features are then fed to a Bi-Directional Long Short-Term Memory (Bi-LSTM) network. The Bi-LSTM is particularly well suited for modelling past and future contextual dependencies, which is useful for discovering subtle and sequential anomaly patterns in cloud environments.

The output of Bi-LSTM layer is passed to a fully connected layer followed by a SoftMax activation and an output layer for generating the final predictions. The hybrid model of densely connected MHA-based AlexNet and Bi-LSTM makes it possible for the DMHA-ABL model to learn effectively rich spatial-temporal representations and ensure high accuracy on cloud intrusion detection tasks.

3.2.1 Multi-Head Attention

MHA is introduced at the first step of the convolutional pipeline to enhance the ability of the model to recognize intricate and diverse spatial relationships among cloud traffic data. MHA enables the architecture to attend to various representational subspaces simultaneously, allowing the model to learn from various patterns in parallel. This is particularly beneficial for intrusion detection tasks, where subtle, intricate, and distributed anomalies can manifest differently across heterogeneous traffic segments.

$$MultiHead(X,Y) = \underset{i \in [N_h]}{concat}\left[H^{(i)}\right]W_O \tag{4}$$

where $H^{(i)}$ represents the attention output of the i-th head, computed as:

$$H^{(i)} = Attention\left(XW_Q^{(i)}, YW_K^{(i)}, YW_V^{(i)}\right) \tag{5}$$

Here, each head has its own set of learnable projection matrices: $W_K^{(i)} \in R^{D_{in} \times d_k}$, $W_K^{(i)} \in R^{D_{in} \times d_k}$, $W_V^{(i)} \in R^{D_{in} \times d_k}$. The output from all N_h heads are concatenated and then projected using the output transformation matrix $W^O \in R^{D_{in} \times d_k}$ to match the desired output dimension D_{out}.

In this setting, d_k refers to the dimensionality of each head's query/key representation, and the total attention dimensionality becomes $D_k = N_k \cdot d_k$. By leveraging the use of multiple heads, the MHA mechanism in DMHA-ABL allows the model to learn a

richer and more intricate description of spatial relations in the cloud data, making its sensitivity to diverse intrusion signatures stronger. This enrichment assists the earlier layers of the network to learn highly discriminative features, thus assisting in more accurate and robust anomaly detection.

3.2.2 Bi-LSTM

The Bidirectional Long Short-Term Memory (BiLSTM) layer is vital to capturing the temporal and contextual dependencies in cloud traffic sequences. In contrast to normal LSTMs that operate information in a one-way temporal direction, BiLSTM layer has two separate LSTM networks, i.e., forward and backward, which link to the same input and deliver the results to the next layer. All input terms pass from $d = 1\ to\ d = N$ in the forward direction, while the opposite is passed from $d = N\ to\ d = 1$ in the reverse direction. Each LSTM unit consists of a cell, input, output, and forget gates. The three gates control the information flow through the cell, which holds the network's temporal state. The LSTM processes the input and uses the following equations to gather the related information.

$$inp_d = \sigma\left(\omega_{inp}[r_d, hid_{d-1}] + B_{inp}\right) \quad (6)$$

$$\mathcal{F}_d = \sigma(\omega_{\mathcal{F}}[r_d, hid_{d-1}] + B_{\mathcal{F}}) \quad (7)$$

$$\mathbb{C}_d = \mathcal{F}_d \cdot \mathbb{C}_{d-1} + inp_d \cdot tanh(\omega[r_d, hid_{d-1}] + B_{\mathbb{C}}) \quad (8)$$

$$out_d = \sigma(\omega_{out}[r_d, hid_{d-1}] + B_{out}) \quad (9)$$

$$hid_d = out_d \cdot tanh(\mathbb{C}_d) \quad (10)$$

The logistic sigmoid function is represented by the symbol σ, the elementwise product is denoted by $\cdot$, the hidden state vector is represented by hid_d, the cell state vector is represented by $\mathbb{C}_d$, the input gate, forget gate, and output gates are represented by inp_d, $\mathcal{F}_d$, and out_d, respectively, and B_{inp}, $B_{\mathcal{F}}$, $B_{\mathbb{C}}$, and B_{out} are the bias vectors, while ω_{inp}, $\omega_{\mathcal{F}}$, and ω_{out} stand for the input gate, forget gate, and output gate's respective weight vectors. Finally, the following equations display the output of the dth term in the BiLSTM layer.

$$F_{hid_d} = \alpha_F(\omega_d, hid_{d-1}), d \in [1, N] \quad (11)$$

$$B_{hid_d} = \alpha_B(\omega_d, hid_{d+1}), d \in [1, N] \quad (12)$$

$$hid_d = [F_{hid_d} \oplus B_{hid_d}] \quad (13)$$

where α_F and α_B are the forward and backward LSTM functions, respectively, and $\oplus$ is concatenation. The BiLSTM thereby achieves bidirectional context, enhancing the model's capability to discriminate between normal and malicious activity even if patterns are distributed over time or appear with delayed correlation.

After the BiLSTM layer, the temporally extracted features are propagated through a fully connected (dense) layer so that the obtained high-level learned representations are transformed to a more classification-friendly discriminative feature space. Finally, the output is sent through a SoftMax activation function to compute the probability distribution across intrusion classes. This enables the model to classify varied categories of attacks, such as DDoS, malware propagation, and unauthorized access, with confidence. SoftMax and the dense layers constitute the decision-making process, mapping learned spatio-temporal features into accurate and interpretable predictions and completing the end-to-end intrusion detection pipeline.

For further optimizing the performance of the model, the Adaptive Whale Optimization Algorithm (AWOA) optimizes key hyperparameters like epoch count, batch size, and learning rate for improved model performance. AWOA dynamically maintains exploration and exploitation in the hyperparameter space to enable the model to converge more quickly, generalize significantly better, and have better detection accuracy for various and intricate intrusion patterns. This optimization approach enhances the overall intrusion detection system robustness and efficiency.

3.3 Adaptive Whale Optimization Algorithm (A-WOA)

The objective function used in the optimization procedure of our proposed DMHA-ABL architecture is the binary cross-entropy (BCE) loss, which is an effective measure of error between predicted probabilities and true binary class labels. The BCE loss is mathematically defined as:

$$J_{bce} = -\frac{1}{M}\sum_{m=1}^{M}\left[y_m \times log\left(h\theta(x_m)\right) + (1 - y_m) \times log\left(1 - h\theta(x_m)\right)\right] \quad (14)$$

where M represents the total number of training samples in the dataset. Each training sample is characterized by two components: y_m which denotes the target label corresponding to the training example m, and x_m which represents the input features for that example. h_θ corresponds to the prediction generated by the DMHA-ABL model parameterized by weights θ.

While training, the hyperparameters of DMHA-ABL, like weights, learning rate, batch size, and epoch are adjusted optimally by the AWOA. The objective of optimization here is to minimize the BCE loss for all training samples so that improved prediction accuracy and better generalization of the model on unseen data are achieved.

The three tactics used by the WOA to mimic whale behavior include spiral bubble-net attack (exploitation phase), hunting for prey (exploration stage), and encircling the prey.

At iteration t, the position of the ith whale is given by,$X_i^t = \left(x_{i,1}^t, x_{i,2}^t, \ldots\ldots x_{i,D}^t\right)$ where $i = 1, 2, \ldots, N$ and $N\ and\ D$. The following subsections provide a mathematical presentation of WOA techniques.

Step 1: Encircling Prey Strategy

Whales can locate and encircle their prey. Whales in WOA consider their intended victim or a place close to it inside the area of search to be the best option. Other whales attempt to approach the optimal agent during prey encirclement and use Equation (1) to update their location. This equation's t is the current iteration X_i^t is the location of the ith whale in the present iteration, and X^{*t} is the position vector of the greatest answer so far, which is modified in every iteration if a better solution is found.

$$X_i^{t+1} = X^{*t} - A.D \tag{15}$$

$$D = |C \times X^{*t} - X_i^t| \tag{16}$$

where D is the distance, as determined by Eq. (2), between the whale, X_i^t and the prey, X^{*t} where symbols $|.|$ represent the absolute value, and $A\ and\ C$ are coefficient matrices that are determined by Eqs. (9) and (10)

$$A = 2 \times a \times r - a \tag{17}$$

$$C = 2 \times r \tag{18}$$

$$a = 2 - t \times \frac{2}{MaxIter} \tag{19}$$

The parameter r in Eqs. (17) and (18) is a random value between 0 and 1, and Eq. (19) states that an in Eq. (17) decreases linearly from 2 to 0. The present iteration along with the total amount of repetitions are indicated by the variables t & $MaxIter$ in Eq. (19). The whales are gradually confined to the surrounding scope by the parameter a.

Step 2: Bubble-Net Attacking Strategy

The bubble-net hunting mechanism in the Whale Optimization Algorithm (WOA) imitates the humpback whales' cooperative hunting strategy and is divided into two behaviors: the spiral updates positioning and the diminishing encircling technique. The spiral motion is mathematically modelled as:

$$X(t+1) = \overrightarrow{D'} \cdot e^{bl} \cos(2\pi l) + X^*(t) \tag{20}$$

$$\overrightarrow{D'} = |X^*(t) - X(t)| \tag{21}$$

The conventional spiral updating method in WOA constrains the exploration ability and has the risk of premature convergence. To solve this, we modify the final position update equation by adding a rank-based inertia weight parameter ω, which adapts exploration and exploitation dynamically according to solution quality. The new equation is:

$$X(t+1) = \begin{cases} X^*(t) - A \cdot D \; if \; p < 0.5 \\ \omega \cdot \left(\overrightarrow{D'} \cdot e^{bl} \cos(2\pi l)\right) + X^*(t) \; if \; p \geq 0.5 \end{cases} \quad (22)$$

Where the inertia weight ω for each agent at time t is defined as:

$$\omega_i(t) = \omega_{min} + \frac{R_i(t)}{ns}(\omega_{max} - \omega_{min}) \quad (23)$$

Here, $R_i(t)$ is the fitness rank of the $i-th$ whale at iteration t, ns is the total number of search agents, and ω_{min}, ω_{max} are user-defined lower and upper bounds for the inertia weight. By assigning less inertial weights to fitter solutions and greater weights to less-fitting ones, this adaptive method provokes more intensive exploitation near hopeful solutions while ensuring a broad exploration for the less-fit agents, promoting convergence speed and avoiding local optima.

Step 3: Searching for Prey Strategy

In the search of a prey strategy, whales investigate the search space to enhance diversity by updating their positions relative to a randomly chosen whale using:

$$X(t+1) = X_{rand} - A \cdot D \quad (24)$$

$$D = |C \cdot X_{rand} - X(t)| \quad (25)$$

where X_{rand} is a random position, and A and C are control parameters. WOA alternates between the position of spiral update ($p \geq 0.5$), encircling prey ($p < 0.5, |A| < 1$), and searching for prey ($p < 0.5, |A| \geq 1$) strategies. The best solution is iteratively refined until maximum iterations are reached.

4 RESULTS AND DISCUSSION

This section illustrates the effectiveness of the proposed DMHA-ABL in intelligent intrusion detection within cloud computing. The model is thoroughly tested in terms of performance using a variety of key metrics such as Accuracy, Precision, Recall, F1-Score, Specificity, Negative Predictive Value (NPV), False Positive Rate (FPR), False Negative Rate (FNR), Matthews Correlation Coefficient (MCC), and Area Under the ROC Curve (AUC-ROC). Empirical results reveal that the DMHA-ABL model performs better than some current state-of-the-art models like CNN, LSTM, BiLSTM, AlexNet, and Transformer.

4.1 Evaluation Setup

The evaluation setup for the proposed DMHA-ABL framework for anomoly detection was conducted on a system with the following hardware configuration: an Intel Core i5 or i7 processor to ensure efficient computational performance. The system featured a minimum of 8GB RAM, with 16GB or more recommended for managing large datasets and deep learning tasks effectively.

4.2 Dataset Description

NSL-KDD dataset https://www.kaggle.com/code/eneskosar19/intrusion-detection-system-nsl-kdd is a refined form of the KDD Cup 1999 dataset, specifically with the aim to counter its built-in limitations such as class imbalance and duplicate records that might negatively impact the performance of machine learning algorithms. It contains a collection of network traffic data with one of several attack types or normal. The database consists of two prevalent subsets: the training set containing a variety of attack types such as DoS (Denial of Service), U2R (User to Root), R2L (Remote to Local), and Probe, and normal network traffic information; and the test set, which is better balanced with less duplicate instances. The NSL-KDD dataset contains both continuous and categorical features, including network traffic information like duration, protocol type, service, flag, number of bytes, and connection status, making it a great dataset to train machine learning models to be used in intrusion detection applications. The dataset is extensively used to test the accuracy, precision, recall, and overall detection ability of IDS systems in various attack scenarios.

4.2 Performance Evaluation

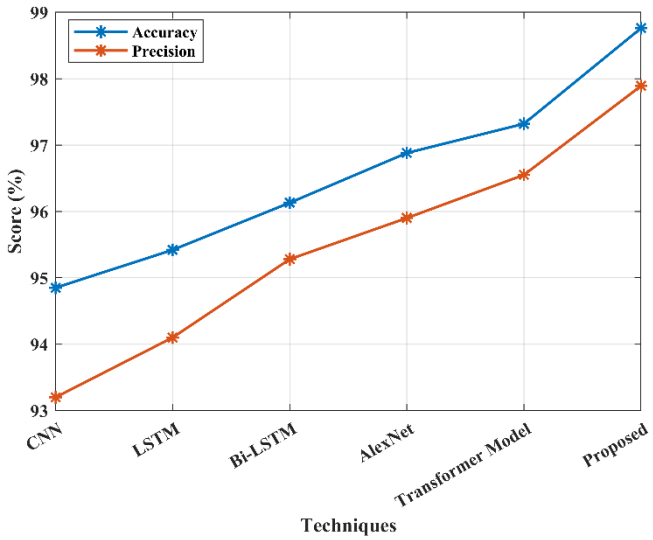
In this assessment, we compare the performance of certain recent deep models, including CNN, LSTM, BiLSTM, AlexNet, and Transformer, with the suggested DMHA-ABL model for intelligent intrusion detection in cloud computing environments. The performance is being compared based on a broad array of performance measures, including Accuracy, Precision, Recall, F1-Score, Specificity, NPV, False Positive Rate (FPR), FNR, MCC, and AUC-ROC. Table 2 shows the performance comparison of a proposed and existing methods.

Table 2 Evaluation of performance comparison with proposed and existing methods

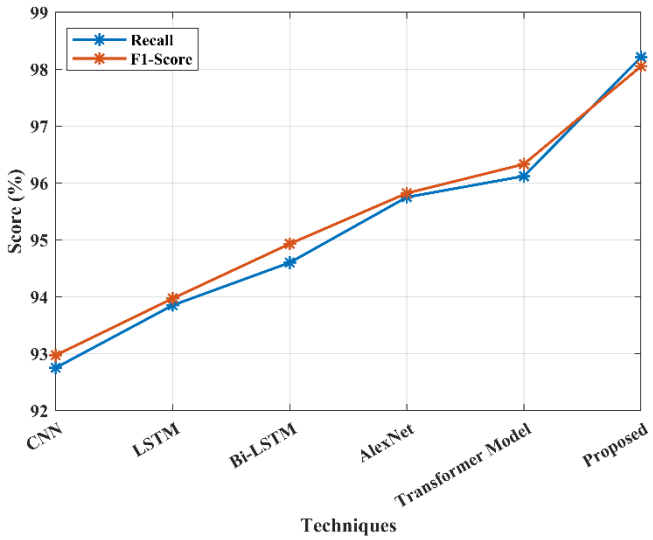
Model	CNN	LSTM	BiLSTM	AlexNet	Transformer	Proposed
Accuracy	94.85	95.42	96.13	96.88	97.32	98.76
Precision	93.2	94.1	95.28	95.9	96.55	97.89
Recall	92.75	93.85	94.6	95.75	96.12	98.21
F1-Score	92.97	93.97	94.93	95.82	96.33	98.05
Specificity	96.15	96.6	97.12	97.55	98.1	99.02

NPV	95.22	95.74	96.3	96.85	97.42	98.85
FPR	3.1	2.84	2.1	1.85	1.45	0.98
FNR	7.25	6.15	5.4	4.25	3.88	1.79
MCC	0.894	0.905	0.917	0.928	0.936	0.961
AUC-ROC	0.965	0.972	0.978	0.981	0.987	0.992

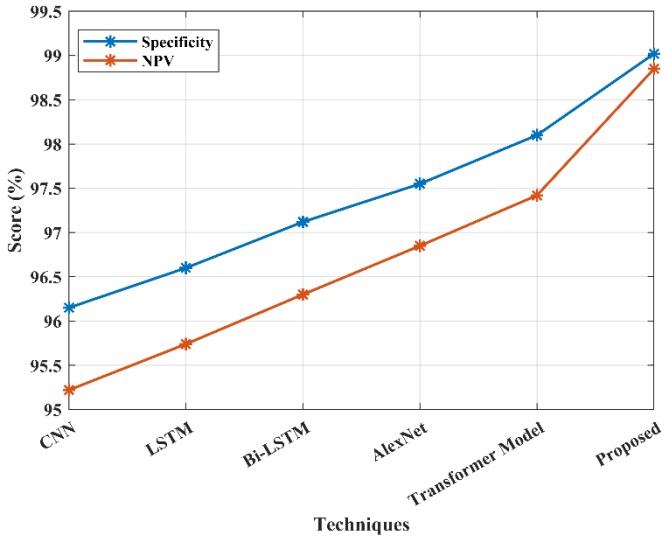
Table 2 illustrates a comparative assessment of the performance of the new model against a number of prevalent deep learning architectures, i.e., CNN, LSTM, BiLSTM, AlexNet, and Transformer. On all performance metrics, recall, accuracy, NPV, precision, F1-score, specificity, FNR, FPR, MCC, and AUC-ROC the new model outperforms the rest on all accounts. It boasts the best 98.76% accuracy, outperforming the 97.32% of the Transformer model, and showcases better precision (97.89%), recall (98.21%), and F1-score (98.05%), reflecting strong predictiveness and equitable classification. Additionally, the introduced model reflects superior robustness with the lowest FPR (0.98%) and FNR (1.79%), and highest specificity (99.02%) and NPV (98.85%), reflecting its efficiency in reducing false alarms and missed detections. Additionally, the MCC value of 0.961 and the AUC-ROC value of 0.992 signify the high predicted-to-actual class correlation of the model as well as its superb overall classification performance. These results confirm the excellence of the proposed model in correctly classifying intrusions with high reliability and stability for all performance measures.



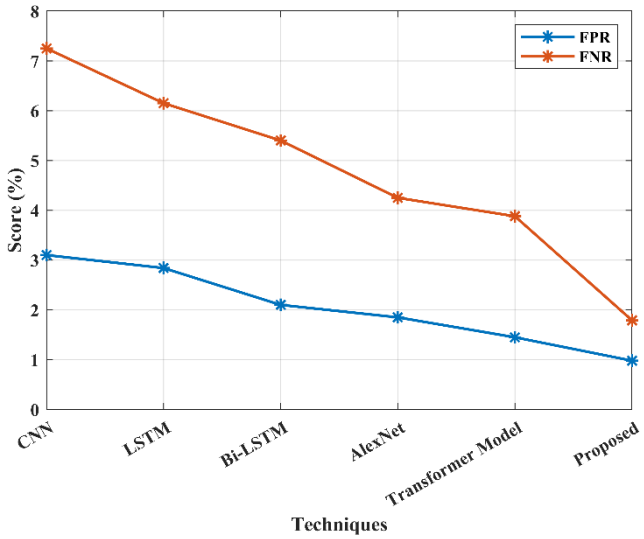
(i) Accuracy, Precision



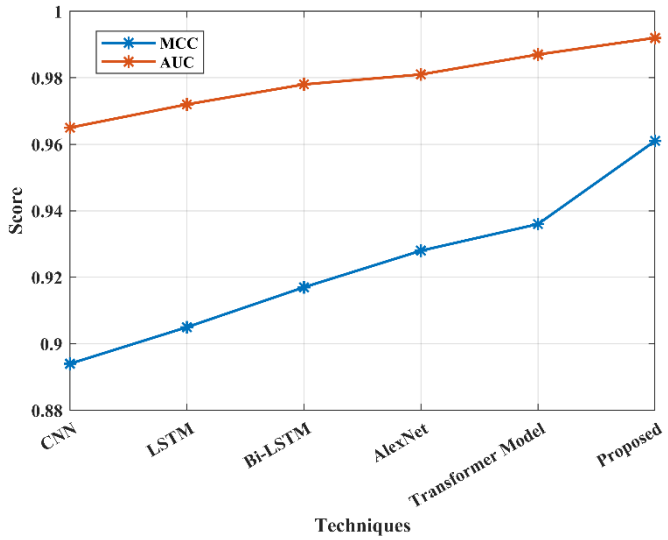
(ii) Recall, F1-Score



(iii) Specificity, NPV



(iv) FPR, FNR



(v) MCC, AUC

Figure 3 (i)-(v) Visual representation of several performance metrics with proposed and existing works

The performance metrics, which include sensitivity, accuracy, specificity, NPV, precision, FNR, FPR, MCC, F-Measure, AUC, and Specificity are represented graphically in Figures 3 (i) to (v). These metrics are compared with several approaches, including Proposed, CNN, LSTM, Bi-LSTM, AlexNet, Transformer model.

4.3 Hyperparameter Tuning

Hyperparameter tuning is an essential task for optimizing the performance of the given model. We tested some different hyperparameters such as the number of epochs, batch size, and learning rate to optimize the model for optimal performance. Grid search and adaptive optimization methods were used to tune in order to find the optimal setting that gives the best precision, recall, accuracy, and other metrics. Results indicate that

optimization of these hyperparameters significantly improves convergence rate, detection rate, as well as global stability of the model for the identification of intrusion attempts. Such results highlight the importance of hyperparameter tuning towards achieving high-performance intrusion detection systems. Table 3 shows the tuning results.

Table 3 Results for Hyperparameter tuning

Epoch	Batch Size	Learning Rate	Accuracy	Precision	Recall	F1-Score
20	32	0.01	93.45	92.12	91.67	91.89
40	32	0.005	95.12	94.05	93.88	93.96
60	64	0.005	96.84	96.11	95.92	96.01
80	64	0.001	97.93	97.22	97.01	97.11
100	128	0.0005	98.76	97.89	98.21	98.05

Table 3 illustrates the hyperparameter tuning outcome of the model proposed in this research by adjusting epochs, batch size, and learning rate. It is clearly noticeable in the outcomes that model performance continues to increase with increasing epochs in training along with optimized hyperparameters. With 20 epochs, learning rate 0.01, and batch size 32, the accuracy of model was 93.45%, precision was 92.12%, recall was 91.67%, and F1-score was 91.89%. With the training boosted up to 40 and 60 epochs and with lower learning rates and bigger batch size, the performance got significantly better. Especially when a batch size was 64, the learning rate was 0.001 and the number of epochs was 80, model accuracy was 97.93%, showing highly good generalization ability. Finally, the most optimal performance appeared at 100 epochs when there was a batch size of 128 and optimally tuned learning rate of 0.0005, which had accuracy (98.76%), precision (97.89%), recall (98.21%), and F1-score (98.05%). These results confirm that accurate calibration of training parameters greatly enhances the model's performance and stability.

5 CONCLUSION

The suggested DMHA-ABL framework demonstrates an effective and intelligent intrusion detection method specifically tailored for cloud computing platforms. Through the integration of a synergistic preprocessing method with a hybrid deep learning model consisting of densely connected Multi-Head Attention-augmented AlexNet and BiLSTM the model efficiently extracts spatial and temporal features crucial for identifying sophisticated

and dynamic cyber-attacks. In addition, AWOA combined with rank-based inertia weight adaptation enhances training significantly by fine-tuning significant hyperparameters appropriately, providing improved convergence as well as detection rates. Experimental results affirm the reasonableness, scalability, and overall improved performance of the system as suggested in attack detection against broad ranges of attacks such as DDoS, unauthorized access, as well as spreading malware. The model achieved an optimum accuracy of 98.76%, precision of 97.89%, recall of 98.21%, and F1-score of 98.05%, better than state-of-the-art methods. DMHA-ABL thus presents an efficient and scalable scheme to secure dynamic cloud infrastructures. In the future, the model can be made to accept detection with minimum latency and with the integration of blockchain for guaranteed data reliability and trust in distributed clouds.

REFERENCE

1. Nzeako, R.A.S.G. and Shittu, R.A., 2024. Leveraging AI for enhanced identity and access management in cloud-based systems to advance user authentication and access control. *World Journal of Advanced Research and Reviews*, *24*(3), pp.1661-1674.
2. Dey, S., Sarma, W. and Tiwari, S., 2023. Deep learning applications for real-time cybersecurity threat analysis in distributed cloud systems. *World Journal of Advanced Research and Reviews*, *17*(3), pp.1044-1058.
3. Torre, D., Chennamaneni, A., Jo, J., Vyas, G. and Sabrsula, B., 2025. Toward Enhancing Privacy Preservation of a Federated Learning CNN Intrusion Detection System in IoT: Method and Empirical Study. *ACM Transactions on Software Engineering and Methodology*, *34*(2), pp.1-48.
4. Abdullayeva, F., 2023. Cyber resilience and cyber security issues of intelligent cloud computing systems. *Results in Control and Optimization*, *12*, p.100268.
5. Hnamte, V. and Hussain, J., 2023. Dependable intrusion detection system using deep convolutional neural network: A novel framework and performance evaluation approach. *Telematics and Informatics Reports*, *11*, p.100077.
6. Oladosu, S.A., Ige, A.B., Ike, C.C., Adepoju, P.A., Amoo, O.O. and Afolabi, A.I., 2022. Revolutionizing data center security: Conceptualizing a unified security framework for hybrid and multi-cloud data centers. *Open Access Research Journal of Science and Technology*, *5*(2), pp.086-076.
7. Lukaj, V., 2023. Physical and cyber systems for smart environments: IoT and Digital Twins.

8. Carlin, A., Hammoudeh, M. and Aldabbas, O., 2015. Intrusion detection and countermeasure of virtual cloud systems-state of the art and current challenges. *International Journal of Advanced Computer Science and Applications*, *6*(6).
9. AlSaleh, I., Al-Samawi, A. and Nissirat, L., 2024. Novel machine learning approach for DDoS cloud detection: bayesian-based CNN and data fusion enhancements. *Sensors*, *24*(5), p.1418.
10. Mohammed, S.H., Singh, M.S.J., Al-Jumaily, A., Islam, M.T., Islam, M.S., Alenezi, A.M. and Soliman, M.S., 2025. Dual-hybrid intrusion detection system to detect False Data Injection in smart grids. *PloS one*, *20*(1), p.e0316536
11. Hariharan, S., Jerusha, Y.A., Suganeshwari, G., Ibrahim, S.S., Tupakula, U. and Varadharajan, V., 2025. A Hybrid Deep Learning Model for Network Intrusion Detection System using Seq2Seq and ConvLSTM-Subnets. *IEEE Access*.
12. Pushpa, B.R., Jyothsna, S. and Lasya, S., 2025. HybNet: A hybrid deep models for medicinal plant species identification. *MethodsX*, *14*, p.103126.
13. Afyouni, I., Khan, A. and Aghbari, Z.A., 2022. Deep-Eware: spatio-temporal social event detection using a hybrid learning model. *Journal of big Data*, *9*(1), p.86.
14. Abdallah, A.M., Alkaabi, A.S.R.O., Alameri, G.B.N.D., Rafique, S.H., Musa, N.S. and Murugan, T., 2024. Cloud network anomaly detection using machine and deep learning techniques—Recent research advancements. *IEEE Access*, *12*, pp.56749-56773.
15. Chauke, K.O., Muchenje, T. and Makondo, N., 2025. Enhancing Network Security in Multi-Cloud Environments through Adaptive Threat Detection.
16. Bahaa, A., Sayed, A., Elfangary, L. and Fahmy, H., 2022. A novel hybrid optimization enabled robust CNN algorithm for an IoT network intrusion detection approach. *Plos one*, *17*(12), p.e0278493.
17. Aljuaid, W.A.H. and Alshamrani, S.S., 2024. A deep learning approach for intrusion detection systems in cloud computing environments. *Applied Sciences*, *14*(13), p.5381.
18. Najafli, S., Toroghi Haghighat, A. and Karasfi, B., 2024. A novel reinforcement learning-based hybrid intrusion detection system on fog-to-cloud computing. *The Journal of Supercomputing*, *80*(18), pp.26088-26110.
19. Chiba, Z., Alaoui, M.E.K., Abghour, N. and Moussaid, K., 2022. Automatic building of a powerful IDS for the cloud based on deep neural network by using a novel

combination of simulated annealing algorithm and improved self-adaptive genetic algorithm. *International Journal of Communication Networks and Information Security*, *14*(1), pp.93-117.

20. Long, Z., Yan, H., Shen, G., Zhang, X., He, H. and Cheng, L., 2024. A Transformer-based network intrusion detection approach for cloud security. *Journal of Cloud Computing*, *13*(1), p.5.
21. Sorour, S.E., Aljaafari, M., Shaker, A.M. and Amin, A.E., 2025. LSTM-JSO framework for privacy preserving adaptive intrusion detection in federated IoT networks. *Scientific Reports*, *15*(1), p.11321.
22. Veeramachaneni, V., 2025. Adaptive Ensemble Learning Framework for Robust Intrusion Detection in Wireless Sensor Networks. *Journal of Advancement in Parallel Computing*, *8*(1), pp.18-31.
23. Bukhari, S.M.S., Zafar, M.H., Abou Houran, M., Moosavi, S.K.R., Mansoor, M., Muaaz, M. and Sanfilippo, F., 2024. Secure and privacy-preserving intrusion detection in wireless sensor networks: Federated learning with SCNN-Bi-LSTM for enhanced reliability. *Ad Hoc Networks*, *155*, p.103407.
24. He, H., Li, X., Chen, P., Chen, J., Liu, M. and Wu, L., 2024. Efficiently localizing system anomalies for cloud infrastructures: a novel dynamic graph transformer based parallel framework. *Journal of Cloud Computing*, *13*(1), p.115.
25. Gwassi, O.A.H., Uçan, O.N. and Navarro, E.A., 2024. Cyber-XAI-Block: an end-to-end cyber threat detection & fl-based risk assessment framework for iot enabled smart organization using xai and blockchain technologies. *Multimedia Tools and Applications*, pp.1-42.