International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(19s) 2026 pp. 612-626
https://doi.org/10.70917/ijcisim-2026-5068
Article

A Genetic Algorithm-Optimized Hybrid Deep Learning Framework for Accurate and Robust Intrusion Detection in Cloud Computing

Sabria Ahmed Ben Naser[1], Farij Omer Ehtiba[2], Haitham Saleh Ben Abdelmula3*, Abdulhkim Alhadi Anaiba[4]

1 School of Basic Science, Libyan Academy for Postgraduate Studies-Misurata, Misurata, Libya
2 Faculty of Information Technology, Misurata University, Misurata, Libya
3,4 Cyber Security Department, Libyan Center For Electronic Systems, Programming and Aviation Research, Tripoli, Libya
[1]sabria.bennaser@it.lam.edu.ly, 2f.ehtiba@it.misurata.edu.ly, 3hsaa8383@gmail.com, 4abduhk71@yahoo.com
1 https://orcid.org/0009-0001-9185-2668; 2 https://orcid.org/0009-0000-5820-2156; 3 https://orcid.org/0009-0005-2604-8648; 4 https://orcid.org/0009-0008-3653-3569
*Corresponding Author: hsaa8383@gmail.com

Abstract: The rapid expansion of cloud computing has broadened the cyber threat landscape, making intrusion detection increasingly difficult due to the scale, speed, and complexity of network traffic in dynamic cloud environments. This paper presents A Genetic Algorithm-Optimized Hybrid Deep Learning-driven Intrusion Detection System (GA-HDLIDS) that combines Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, and an attention mechanism to model both spatial and temporal patterns in network data. A Genetic Algorithm (GA) is integrated to automate hyperparameter optimization, eliminating manual tuning and significantly boosting classification accuracy. Evaluated on the CSE-CIC-IDS2018 dataset, the baseline model (HDLIDS) achieves 99.19% (multi-class) and 99.57% (binary) accuracy. The OHDLIDS model substantially outperforms existing methods, reaching 99.97% and 99.99% accuracy, respectively, with F1 scores above 99.9%. Moreover, the model generalizes effectively to an unseen dataset (CSE-CIC-IDS2019) without retraining, attaining 90.91% accuracy, confirming its robustness in real-world scenarios. All experiments prevent data leakage through strict training-testing separation. These results establish OHDLIDS as a highly accurate, scalable, and practical solution for intrusion detection in evolving cloud computing environments.
Keywords: cloud computing, cyber security, deep learning, genetic algorithm, hybrid model

1. Introduction

Over the past two decades, the world has witnessed a radical transformation in the consumption and delivery of digital services, most notably the widespread adoption of cloud computing as a flexible and efficient platform for storing and processing data. According to the National Institute of Standards and Technology (NIST), "cloud computing is a model that provides convenient, on-demand access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be provisioned and released rapidly with minimal management effort or service provider interaction"[1]. Dynamic nature of the cloud to bypass traditional security systems. This makes cloud security a top priority for both users and service providers [2]. With the increasing frequency and sophistication of cyberattacks, cyber security has become a crucial pillar for enhancing the continuity and stability of digital systems, particularly in open and dynamic environments such as cloud computing. Intrusion Detection Systems (IDS) are a fundamental component of cyber defence and form a primary line of defence, monitoring network traffic to detect any suspicious activity and alerting network administrators to malicious activities that may indicate an attempted intrusion or exploitation of security vulnerabilities [2]. However, traditional systems

based on fixed rules or predefined signatures struggle to counter modern attacks or those with dynamically evolving methods. Due to these limitations, the widespread use of artificial intelligence, machine learning, and deep learning techniques has led to more adaptable and intelligently analytical detection systems with enhanced performance [3]. Deep learning-based detection systems have gained increasing attention in recent decades for their ability to handle large-scale and complex network traffic more efficiently, learning feature representations from raw data, while reducing the need for human intervention [4].

According to the literature survey, several research studies have been conducted on the detection-based systems in the presence of AI models and proposed many IDS schemes in order to achieve the cloud security performance.

Farhan et al.'s study [5] combined the use of the BPSO algorithm and deep neural networks to enhance intrusion detection systems, along with feature selection from CSE-CIC-IDS2018 data. Classification using a three-layer neural network achieved 95% accuracy, a very high detection rate, and fewer false alarms compared to traditional algorithms such as SVM and J48. However, the study revealed weaknesses in detecting certain attacks (such as brute-force attacks on FTP) due to its failure to account for data imbalances. The study in [6] developed a penetration detection system based on the CSE-CIC-IDS2018 database, using two different networks: CNNs and LSTMs, after processing and parallelizing the data through remodelling. The results showed that the CNN achieved an accuracy of approximately 98.31% with a reasonable training time (about 6.8 hours), while the LSTM achieved similar accuracy (98.15%) and lower loss, but required a much longer training time (over 41 hours). Furthermore, the study demonstrated the ability of the CNN to perform multi-class classification more quickly, while the LSTM appeared to be more computationally expensive despite its similar performance.

Khan [7] proposed a hybrid model, HCRNNIDS, that integrates CNN and RNN for intrusion detection with improved detection of complex attacks. The research used the CSE-CIC-IDS2018 dataset after pre-processing and solving the data imbalance issue owing to oversampling. The system-attained accuracy of 97.75% with extremely low false alarm rate of 1.4%, outperforming traditional schemes such as LR, DT, and XGB, and CNN-based or LSTM-based systems alone. In [8] Al-Zaghimi and Al-Khudairy proposed two models for intrusion detection in cloud computing environments. The first used a multi-layer back propagation neural network (MLP-BP), and the second used the same architecture but optimized the parameters using a particle swarm optimization (PSO) algorithm. Binary and multi-class classification experiments were conducted. Performance results showed that the MLP-BP model achieved a performance of 98.97% and 98.41% in binary and multi-class classification, respectively, which is significantly higher than the MLP-PSO model, which achieved a performance of 96.25% and 95.32%, but failed to classify rare classes such as intrusion and web. Hewapathirana [9] proposed a two-stage method for comparing intrusion detection methodologies on the CSE-CIC-IDS2018 dataset. Stacked automatic encoders (SAE) were first used with MLP classifiers to identify nonlinear features, and the process achieved an accuracy of approximately 96% in binary classification and 93% in classification. The model was more efficient when dealing with complex attacks such as DoS-SlowHTTPTest and FTP-Bruteforce, but was relatively slower in processing. In the second stage, an Apache Spark-based method was used again, based on PCA, logistic regression, and random forest. The model was faster (0.046 seconds for multi-class classification vs. 0.086 seconds for SAE) and could successfully recognize frequent attacks such as DDoS-HOIC and SSH-Bruteforce, but it was unable to classify uncommon attacks such as Infiltration. Another study [3] that adopted a hybrid model combining hierarchical federal learning and knowledge distillation (KD) to reduce data size and improve efficiency. The model is based on a cloud-edge-device architecture, where a large model is trained in the cloud and the knowledge is transferred to smaller models on the device. It achieved a high accuracy of 98.58% in just six rounds, with significant improvements in communication efficiency, reduced data size, and lower inference time and resource consumption.

Most above studies still suffer from limitations related to its reliance on static models without dynamic hyperparameters optimization. This leads to performance variations depending on the data types used and affects accuracy and generalizability. Furthermore, some models treat data sequences equally, failing to differentiate between aspects most indicative of malicious behavior. These models also lack high generalizability when dealing with new attacks due to their heavy reliance on existing training data. Therefore, by considering the importance of evolutionary

algorithms in this paper, the hybrid intrusion detection system for cloud computing that relies on CNNs and LSTMs with an attention mechanism that enhanced by exploit the characteristics of a genetic algorithm. CNNs extract spatial features, LSTMs address temporal dependencies, and the attention mechanism focuses on the most important features of malicious activity. Whereas the genetic algorithm optimizes parameters to achieve more accurate and flexible performance.

2. models and methods

This section explains the methodology used to design an intrusion detection system in a cloud-computing environment using a hybrid model of deep learning techniques and an attention mechanism. To increase the effectiveness of detection and detection of anomalies in traffic, its hyperparameters were optimized using a genetic algorithm. The system was implemented on a real dataset containing different types of attacks (CSECIC-IDS2018).

2.1 Convolutional Neural Network (CNN)

Convolutional neural networks (CNNs) are among the most important deep learning models. Originally designed to handle array-structured data, they have achieved unprecedented success in handling multidimensional structured data, such as images, time signals, and network data. CNNs rely on a mechanism called convolution, which allows for the extraction of local patterns and features from data through filters or cores. The CNN architecture consists of a set of different types of layers arranged in two main components: a convolutional layer and a collection layer, which extracts features from the input layer, and a fully connected layer [10].

2.2 Long-Short Term Memory (LSTM)

LSTM networks are a class of RNNs specifically designed to address the well-known problems associated with gradient explosion and overcome the limitations of standard RNNs. These networks successfully solve this problem and provide predictions based on time-series data, demonstrating their high efficiency in monitoring long-term dependencies. Thanks to their complex. programming and ability to handle the vanishing gradient phenomenon, LSTM modules are easy to train and have been successfully used in numerous intrusion detection and network security applications [11].

2.3 Attention Mechanism

The attention mechanism works similarly to the human gaze, where it usually scans the input to identify the most important parts, then focuses on them to obtain more detailed information, while reducing the impact of unnecessary information. This mechanism is lightweight, easy to calculate, and can be implemented in parallel and at high speed [12].

2.4 Genetic Algorithm

(GAs) are an evolutionary learning method based on mathematical models of heredity and natural selection.

They are a fundamental approach in evolutionary computing and have played a key role in developing and improving solutions. They have proven effective in numerous applications and have improved feature selection. They have also been used to fine-tune hyperparameters in deep learning models [13].

3. methodology

Fig. 1 illustrates the framework adopted for the design, implementation, and evaluation of the proposed hybrid intrusion detection system (GA-HDLIDS), which is based on deep learning and enhanced by a genetic algorithm, and is specifically designed for cloud computing environments. The methodology follows a structured and logical path that ensures experimental accuracy, computational efficiency, and complete separation between the training and testing phases. All learning procedures were performed exclusively on the training dataset to prevent data leakage.

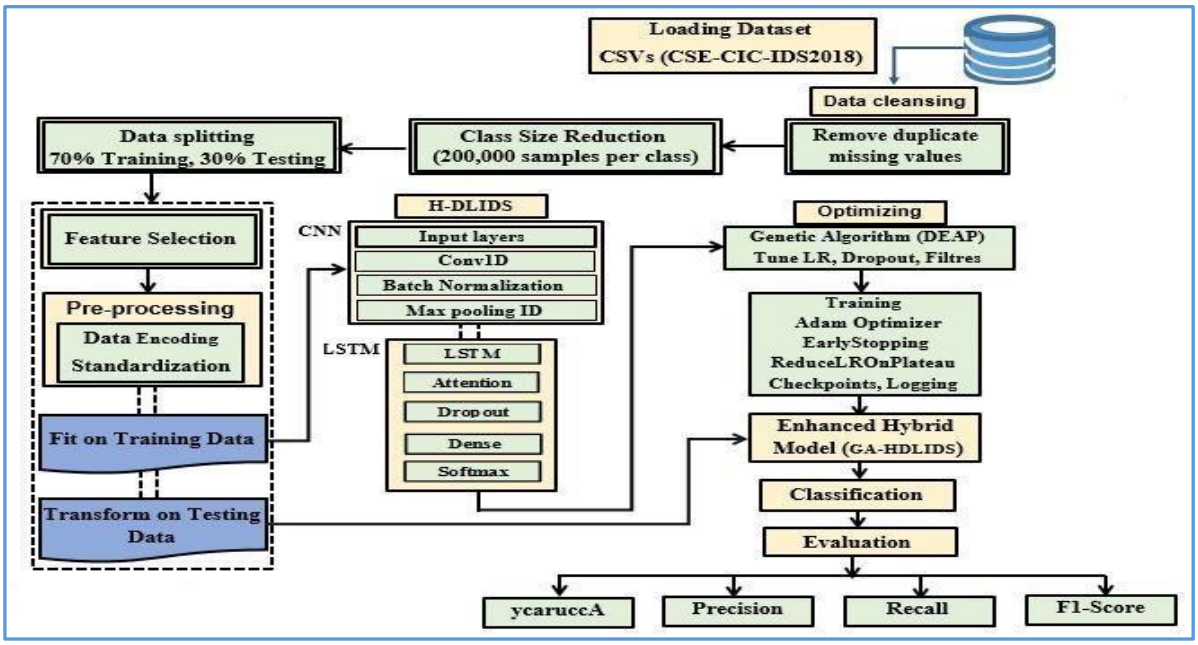

Fig. 1 The proposed GA-HDLIDS model

A. Dataset Description

The CSE-CIC-IDS2018 dataset used in this research is one of the most widely used and up-to-date datasets in the field of cybercrime detection. It was obtained from the Kaggle platform and is available in CSV format, ready for use in training models. The data source is the Canadian Institute for Cyber Security (CIC) at the University of New Brunswick (UNB). The version used in this work contains approximately 16,232,943 network traffic records, each with 80 attributes, and the total size of the CSV data is approximately 6.4 GB, distributed across 10 files. It includes seven main types of attacks. Attacks represent approximately 17% of the total records, or about 2.76 million attack records, while the remaining records represent normal (non-malicious) traffic [14]. Fig.2 illustrates the types of categories and their distribution within the dataset.

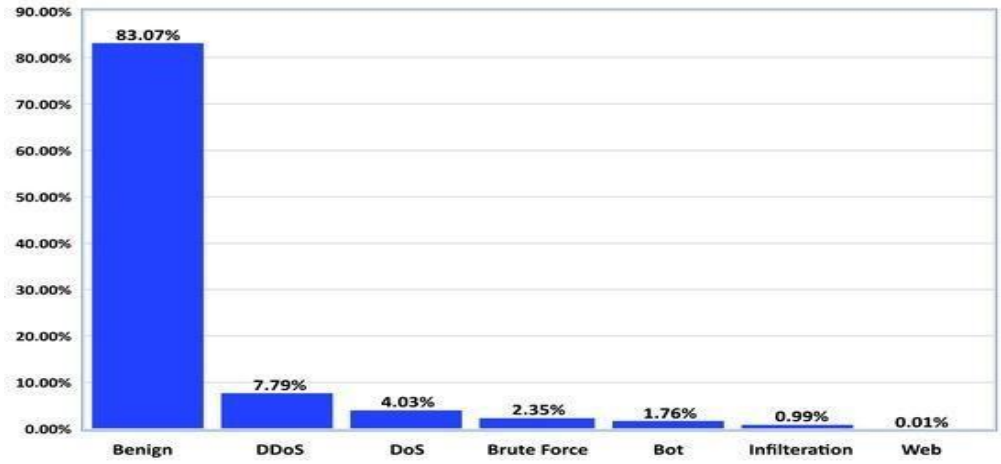

Fig. 2 Distribution of categories in the dataset

B. Pre-processing

Data preprocessing was performed to ensure its quality and consistency, keeping it clean, organized, and usable, through a series of steps. These steps are essential for improving the quality of results by optimizing input quality to guarantee output quality. The data was processed through the following steps:

- Data Cleanup and Preparation: All features were converted to a numerical format to ensure compatibility with machine learning algorithms. Any non-transformable values were treated as missing and removed. Infinite values were replaced with large finite values to prevent numerical instability during model training. Duplicate records were also deleted. In addition, logically inconsistent network flow records that could introduce confusion into the machine learning process were removed. This cleanup phase ensured that the dataset used for modeling consisted only of meaningful and consistent traffic records.
- Filtering Attacks Based on Sample Size: The original dataset showed significant skewness in the distribution of categories, with some categories, such as SQL injection and penetration testing, having extremely low percentages (less than 0.01%). Training deep learning models on these very rare categories could lead to unstable learning behavior and poor convergence. To address this issue, extremely rare attack categories were excluded from the analysis. Furthermore, similar attack types were grouped into uniform categories to enhance category consistency and reduce fragmentation, following preprocessing strategies adopted in previous studies [15]-[16]-[17]. In this study, the categories shown in Table I were exposed, and the RUS method was applied to uniformize each remaining category to encompass 200,000 samples. This resulted in a balanced dataset suitable for training deep learning models. This strategy helped reduce the dominance of common attack categories and achieve a more balanced data distribution, enabling the model to learn distinctive attack patterns more effectively without bias towards any particular category or compromising the quality or diversity of the original data. The Table I shows the size of each category before and after balancing.

Table I

Classification before budgeting

Traffic type	Attack Types	Distribution %	Count (Pre)	Count (Post)
Benign	----	83.070	13,291,200	200000
Advanced DDoS	DDOS -HOIC DDOS-LOIC-UDP	7.786	1,245,760	200000
DoS	DoS-GoldenEye DoS-Slowloris DoS-Hulk, DoS-SlowHTTPTest	4.031	644,960	200000
Brute Force	FTP-Brute Force SSH-Brute force	2.347	375,520	200000
Botnet	----	1.763	282,080	200000

- Data Splitting into Training and Testing: After standardizing class sizes, the dataset was divided using a stratified validation strategy. This method involves separating the dataset into two independent subsets: one used for model training and the other for final performance evaluation. Stratified sampling is applied to ensure

that each class is adequately represented in both subsets, maintaining distribution consistency and preventing class bias during training or evaluation. The stratified validation method was chosen due to the large dataset size and the computational demands of training deep learning with improved genetic algorithms. This validation strategy provides a reliable estimate of model generalization while maintaining computational efficiency. To maintain detection reliability, strict separation of training and test data was maintained throughout all post-processing stages to prevent data leakage.

- Feature selection: Before the feature selection process, a set of columns for the model—"Timestamp," "Source IP Address," "Flow ID," "Source Port," "Protocol," "HTTP," and "SimilarHTTP" were manually excluded because they do not directly contribute to attack detection and are not important. We also created two new features: Flow Bytes/s, representing the data transfer rate during a session, and Flow Packets/s, representing the packet traffic density during a flow period. This was done because some attacks, such as DDoS, are characterized by high traffic density compared to normal traffic. To determine the features most strongly associated with the attack category, Pearson's correlation coefficient was used to measure the linear relationship between each numerical feature and the target. This coefficient is a common statistical tool with values ranging from -1 to 1, where a positive number close to +1 indicates a strong correlation, while a negative number indicates an inverse correlation. Based on the absolute correlation values, the top 25 features in terms of correlation strength for the target category were identified. Five temporal characteristics were also included in the set of features selected for use in the LSTM model, given that this type of neural network relies on temporal patterns in the data. A heat map was created showing how each attribute relates to the others, helping to identify redundancies or multiple correlations between attributes. Fig. 3 illustrates the relationship of each attribute to the others.

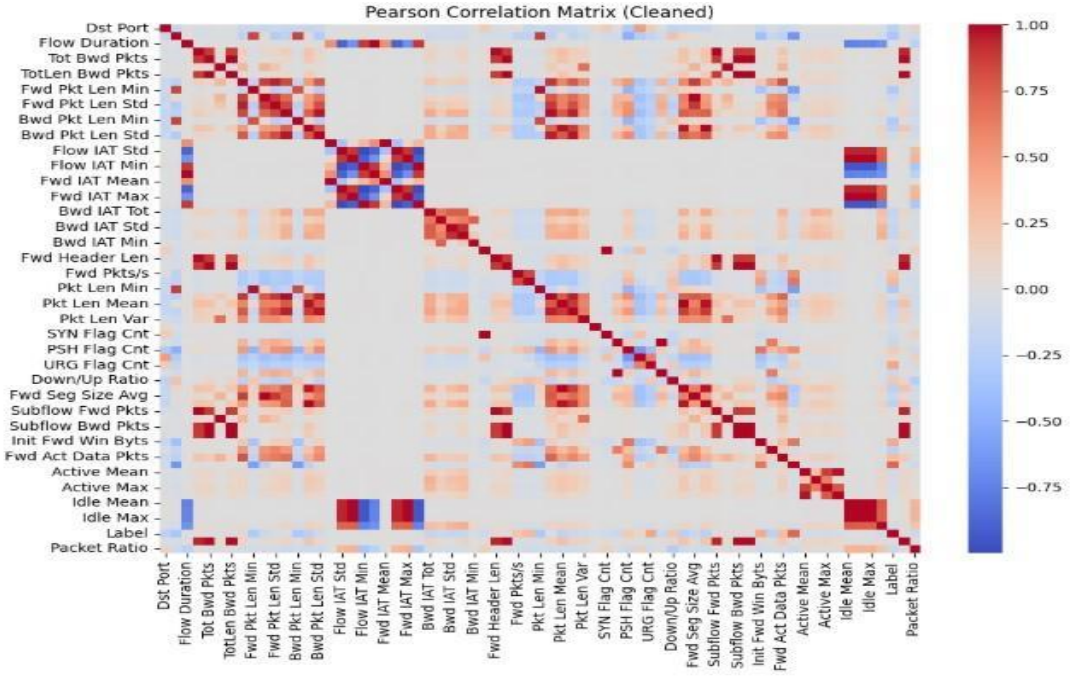

Fig. 3 Heat map showing feature correlation

- Label encoding: Category values in some features were converted to numerical values, enabling deep learning models to understand and process them. For example, fields containing protocol names or session tags were encoded as integers, and the classification column was an object; therefore, it was converted to a computable numerical column, facilitating subsequent statistical analyses that require only numerical data.

- Data standardization: This is a preprocessing operation that readjusts the numerical feature values to conform to a normal distribution with a mean (0) and standard deviation (1). This operation standardizes the range of all features, accelerating convergence and improving the performance of the deep learning model during training. Standardization was applied only to the training data, and the extracted parameters (mean and standard deviation) were used to transform the training and test sets.

C. Hybrid Model Construction (H-DLIDS)

After data processing and preparation, the model was developed based on a hybrid architecture combining CNNs, LSTMs, and an attention mechanism. The model was chosen based on the type of intrusion detection data, which includes a complex spatial and temporal structure. CNNs are used to detect prominent spatial patterns in the data, while LSTMs are suitable for processing network timelines, which in this work represent the dynamic relationship between sequence events in network traffic. The attention mechanism is used in a way that enables the model to focus on the most important features in the timeline more efficiently, improve its predictions, and provide better explanations of the model's behaviour. This integration aims to leverage the strengths of each component in extracting important spatial and temporal features from network traffic data, while enhancing the model's ability to focus on the most significant attack behaviour patterns through the attention mechanism. Fig. 4 illustrates the hybrid model architecture used in the model.

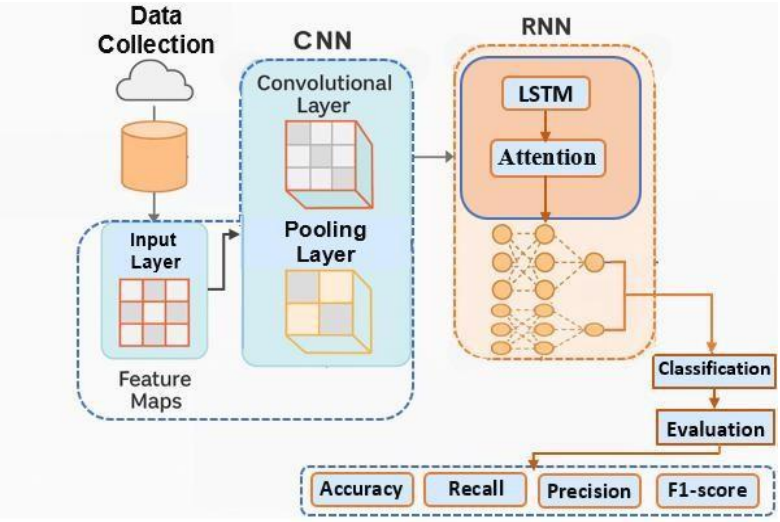

Fig. 4 Hybrid model construction (H-DLIDS)

After building the hybrid model (H-DLIDS), its parameters were optimized using a genetic algorithm based on the principles of natural evolution through selection, mating, and mutation. This algorithm generated multiple sets of parameters in machine learning pathways. To improve its performance by efficiently adjusting the parameters, the optimization process included several key parameters affecting the model's accuracy, such as the number of filters in CNN layer, the number of LSTM layers, the number of nodes in the dense layer, the learning rate, and the batch size. Each possible combination was evaluated using a fitness function, which helped determine the optimal configuration of the model's parameters in a way that improves its accuracy and stability, thus enhancing its effectiveness in detecting cyberattacks in a cloud computing environment. Table II shows the best parameters for the model's performance.

TABLE II
improved model parameters

parameters	Value
Conv1D filters	51

Kernel size	3
Dense Units	148
Dropout Rate	0.435
Learning Rate	0.000590
Epochs	20
Batch size	512
LSTM units	128
Early stopping patience	5
Reduce lr patience	3

D. Training Configuration

The training settings for the proposed hybrid model were modified based on best machine learning practices to ensure efficient and accurate training. The cross-entropy loss function was used to measure the match between the predicted distribution and the actual distribution of the target groups. The Adam algorithm was employed to dynamically optimize learning rates during training, with an early stop mechanism to halt training if performance did not improve within five consecutive periods. The ReduceLROnPlateau mechanism was also used to reduce the learning rate if performance did not improve within three consecutive periods, thus contributing to training stability. Furthermore, a model checkpoint was activated to store the best version based on the highest achieved accuracy, ensuring the most efficient model was used in the evaluation.

4. Results and discussion

This section provides a comprehensive evaluation of the GA-HDLIDS work in a cloud computing environment. The model's performance is analysed using multiple evaluation metrics, including accuracy, fine-tuning, recall, F1 metric, and loss convergence behaviour.

A. Performance Evaluation of the (H-DLIDS) Model

Before genetic optimization of its parameters, the Basic Hybrid Model (H-DLIDS) achieved an overall accuracy of 99.19% on the test dataset. Furthermore, the model achieved high recall and quality values, indicating strong discrimination between normal and malicious data traffic patterns. These results demonstrate that the CNN–LSTM–Attention architecture is highly effective in detecting multi-class intrusions. Despite the strong performance, some minor misclassifications were observed, particularly among structurally similar attack classes. This reflects the similarity in data traffic patterns and the limitations of manually adjusting parameters before evolutionary optimization. Fig. 5 shows the confusion matrix illustrating the performance of the H-DLIDS model, highlighting the classification results across all data traffic classes. Fig. 6 shows the accuracy and loss curves for model training and validation during training cycles.

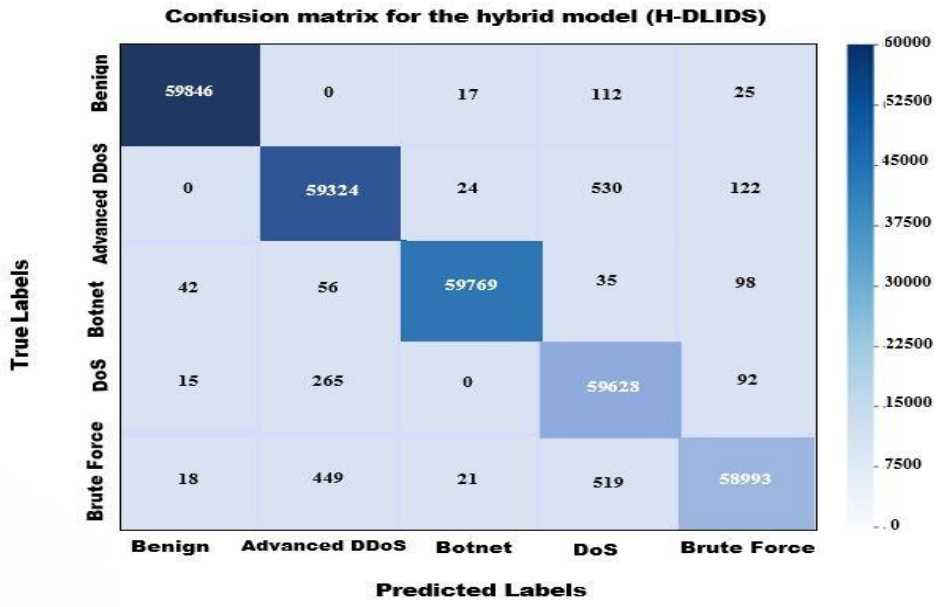

Fig. 5 Confusion matrix for the hybrid model (H-DLIDS)

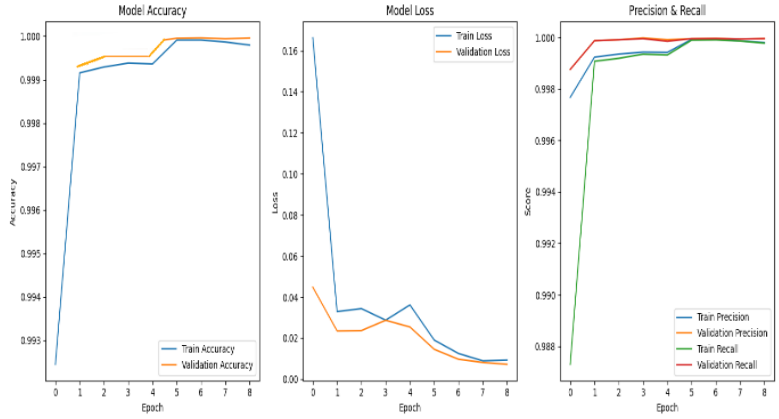

Fig. 6 Accuracy and loss training log for (H-DLIDS)

After testing the model with several manually adjusted hyperparameter settings, the following results demonstrate the best performance achieved by the basic hybrid model (H-DLIDS). As shown in Figures 5 and 6, the confusion matrix in Fig. 5 exhibits high classification capability, with dominant diagonal values in all five categories. Despite this high performance, a limited number of misclassifications were observed. The model produced 154 false alarms (0.26%), where legitimate traffic was misclassified as malicious, and 141 attacks misclassified as legitimate (0.06%), indicating high detection capability. Overlap in the model's classification was also observed between behaviorally similar attack categories, particularly DDoS and DoS attacks, as well as between brute-force and DoS attacks, due to the overlap in traffic characteristics. This overlap suggests that manually adjusting the hyperparameters may not fully optimize the model's decision boundary, necessitating the application of a genetic algorithm to find the optimal parameters required for the model. The detailed evaluation metrics for each class are summarized in Table III.

Table III

Evaluation Performance hybrid model (H-DLIDS)

Class	Precision%	Recall%	F1-score%	Support
Benign	99.91	99.74	99.82	60000
Advanced DDoS	98.88	98.87	98.88	60000
Botnet	99.73	99.62	99.67	60000
Brute Force	98.10	99.32	98.71	60000
DoS	99.35	98.38	98.86	60000
		Accuracy	**99.19**	
Macro avg	99.19	99.19	99.19	300000
Weighted avg	99.19	99.19	99.19	300000

B. Performance Evaluation of the (GA-H-DLIDS) Model

After applying the genetic optimization algorithm, the model's hyperparameters were automatically recalibrated through evolutionary processes including selection, hybridization, and mutation. The optimization process explored a wide range of candidate solutions to determine the optimal parameter configuration. The resulting optimal settings are shown in Table IV. Adjusting these hyperparameters improved the balance between model complexity and generalizability. As a result, the GA-HDLIDS model achieved near-perfect performance across all evaluation metrics, with accuracy, fine-tuning, recall, and F1 score exceeding 99.9% in all experimental scenarios, indicating a significant reduction in classification errors. Fig 7 and 8 illustrate the confusion matrix and accuracy-loss curves for training and validating the GA-HDLIDS model during the training cycles.

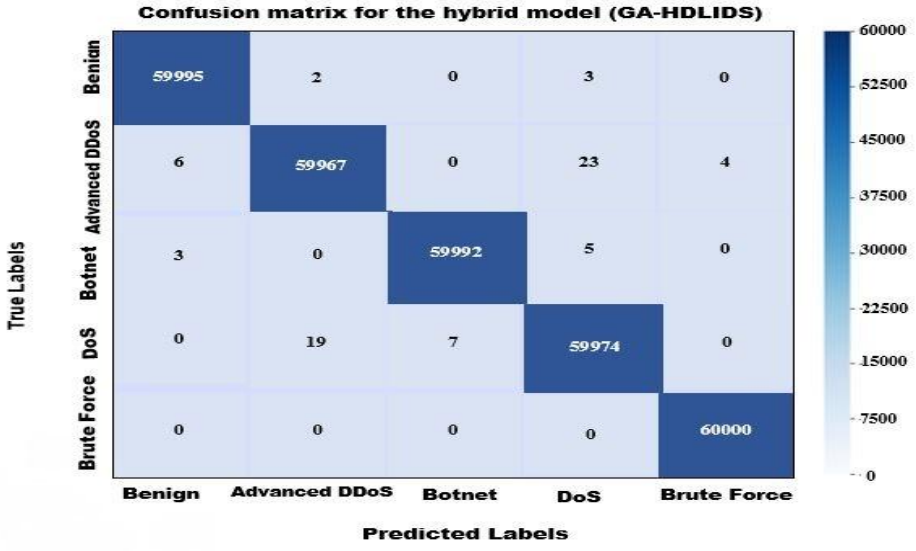

Fig.7 Confusion matrix of model (GA-HDLIDS)

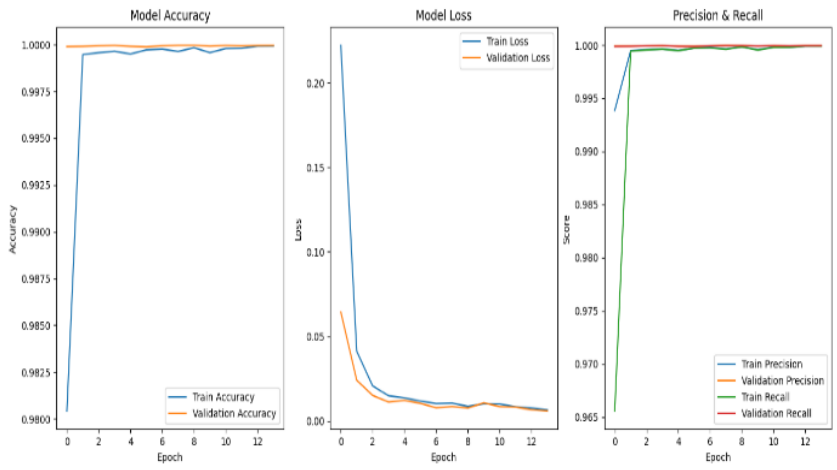

Fig. 8 Accuracy and loss training log model (GA-HDLIDS)

The results show that the improved model achieved an overall accuracy of 99.97%, demonstrating highly reliable classification across all network traffic categories. The overall and weighted averages for accuracy, recall, and F1 score were also approximately 99.97%, confirming that the model maintains balanced performance without bias towards any particular category.

Overall, the GA-HDLIDS model exhibits near-perfect performance across all metrics and categories, with extremely low false classification rates. Table IV illustrates the performance results of the improved model in the final, multi-category evaluation.

Table IV

(GA-HDLIDS) Model Performance Results

Class	**Precision%**	**Recall%**	**F1-score%**	**Support**
Benign	99.98	99.99	99.98	60000
Advanced DDoS	99.96	99.94	99.95	60000
Botnet	99.98	99.98	99.98	60000
Brute Force	99.99	100	99.99	60000
DoS	99.94	99.95	99.95	60000
Accuracy **99.97**				
Macro avg	99.97	99.97	99.97	300000
Weighted avg	99.97	99.97	99.97	300000

C. Performance Comparison

As shown in Table IV, the improved GA-HDLIDS model significantly improved compared to the base H-DLIDS model in both multi-class and binary classification scenarios. Its overall accuracy increased from 99.19% to 99.97%, a rise of approximately 0.78%. While this increase may seem small, it has a substantial impact on intrusion detection systems in cloud computing environments, where massive amounts of data are processed. Furthermore, the number of incorrectly classified samples decreased from 2,440 to 72, representing a 97% reduction in classification errors. This demonstrates that the genetic algorithm helped determine the optimal settings for the hyperparameters,

contributing to a better balance between classification accuracy, retrieval, and the F1 index. This improvement enhances the system's robustness and reliability in protecting cloud infrastructure.

TABLE V

Performance comparison

Hybrid Models		**Accuracy %**	**Precision %**	**Recall %**	**F1-score %**
H-DLIDS	**Multi-class**	99.19	99.29	99.37	99.33
	Binary	99.57	99.74	99.88	99.81
GA-HDLIDS	**Multi-class**	99.97	99.96	99.97	99.97
	Binary	99.99	99.97	99.97	99.97

These results improve the robustness and reliability of the intrusion detection system, which is crucial for protecting sensitive cloud computing infrastructure. Fig 9 illustrates a comparison of the assessment metrics before and after the improvement.

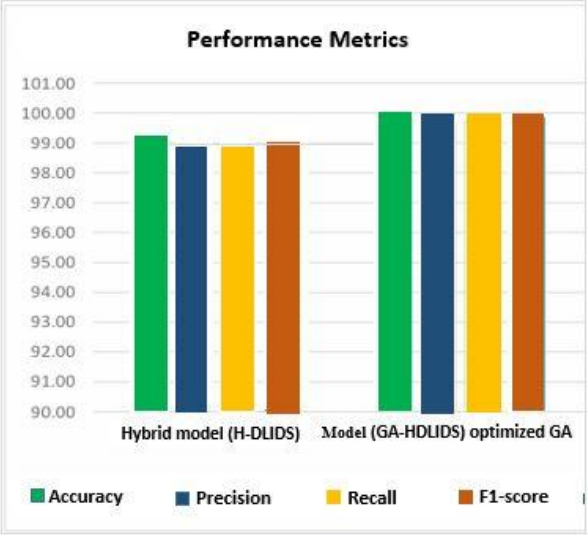

Fig. 9 Comparative analysis of the model results

D. Performance Comparison with Previous Results

As shown in TABLE VI, the proposed system is compared with several recent studies that used CNN-LSTM models for intrusion detection using the CSE-CICIDS2018 dataset. It is important to note that this comparison was conducted within a multi-category classification framework, where the models were required to distinguish between multiple attack categories and normal traffic simultaneously.

The results of the comparison indicate that previous studies achieved high detection performance, with reported detection accuracy ranging from 97.89% to 99.33%. In contrast, the basic hybrid model proposed in this thesis (H-DLIDS) achieved 99.19% accuracy in multi-category assessment. After applying the genetic algorithm optimization (GA-HDLIDS), the performance improved significantly, reaching 99.97% accuracy. These results demonstrate the effectiveness of hyperparameter optimization using genetic algorithms in enhancing attack detection capabilities.

Table VI

Comparison Of GA-HDLIDS with Literature

Dataset	Ref	Year	Learning Algorithm	Accuracy
CSE-CICIDS2018	[18]	2024	CNN-LSTM	97.89%
			FOA-CNN-LSTM	99.33%
	[19]	2024	CNN–LSTM–XGBoost	9.30%
	[20]	2024	CNN-LSTM	99.00%
	[21]	2023	CNN-LSTM	98.85%
	[22]	2024	CNN-LSTM	98.69%
	[23]	2024	CNN-LSTM	98.34%
	Proposed system	2026	CNN-LSTM-Att	99.19%
			CNN-LSTM-Att –GA	99.97%

After adapting the new data, the CSE-CIC-IDS2019 data was evaluated using the GA-HDLIDS model without retraining. A two-way evaluation was performed, simulating real-world environments. The results demonstrated the model's high generalization ability. As shown in the confusion matrix Fig. 10, the model achieved a detection accuracy of 90.91% with a low false alarm rate, demonstrating strong generalization capabilities. The model effectively detected most attacks, especially DDoS and DoS attacks, confirming its ability to handle unfamiliar data traffic patterns.

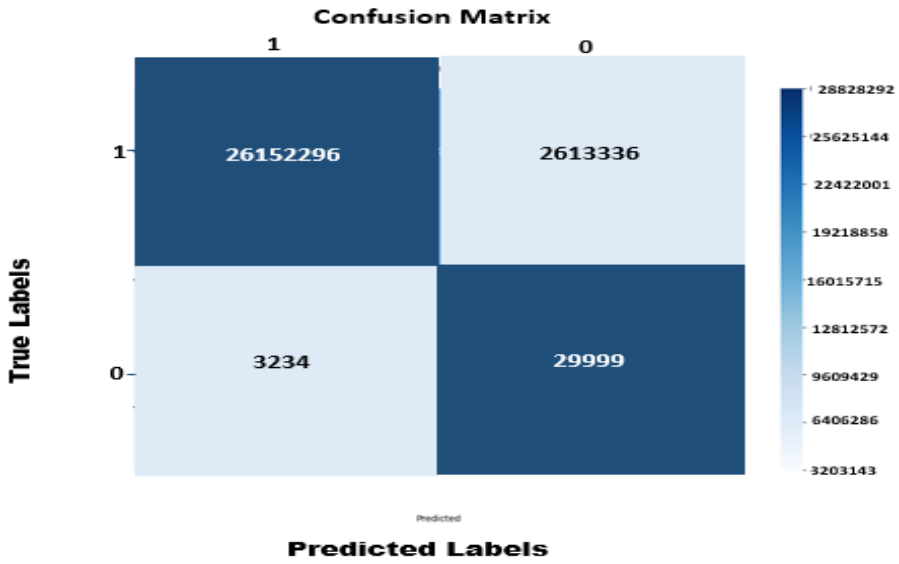

Fig. 10 Confusion matrix of results for new data

TABLE VII

New data evaluation

Class	Precision%	Recall%	F1-score	Support
Benign	1.1	90.3	2.2	33233
Attacks	99.99	90.9	95.2	28765632
Accuracy 90.91				
Macro avg	48.74	90.59	50.56	28798865
Weighted avg	99.87	90.91	99.87	28798865

The CSE-CIC-IDS2019 test results, as depicted in TABLE VII, show that the model maintains stable performance on non-visible data, achieving an accuracy of 90.91%. Weighted metrics indicate strong performance (precision: 99.87%, Recall 90.91%, F1: 99.87%), confirming its robustness in detecting dominant attack traffic. On the other hand, the overall average values appear relatively low. The low overall score is attributed to class imbalances, but the overall results demonstrate effective detection and good generalization to a more recent dataset.

5. Conclusion

This research presents a hybrid intrusion detection system based on deep learning, called GA-HDLIDS, designed to enhance security in cloud computing environments. The proposed model combines (CNNs), (LSTMs), and an attentional mechanism to efficiently capture complex spatial and temporal patterns in network traffic. The primary objective was to optimize results by using a genetic algorithm to automatically find the optimal hyperparameters for the model, thereby increasing detection efficiency and reducing false alarms. The results demonstrated significant improvements and near-perfect accuracy in distinguishing between different attack patterns and reducing false alarms and classification errors. These results confirm the effectiveness of integrating evolutionary optimization with hybrid deep learning models in reducing classification errors and improving detection reliability. The proposed model also demonstrated competitive performance compared to existing CNN-LSTM-based methods. Furthermore, testing the model on the CSE-CIC-IDS2019 invisible dataset confirmed its generalizability to new network environments, achieving stable performance despite class imbalances. The reliability of these results is demonstrated by the adopted evaluation methodology, which prevented data leakage during processing and ensured the model was tested on entirely new data without retraining.

REFERENCES

1. P. Mell and T. Grance, "The NIST definition of cloud computing," 2011.
2. I. Odikayor-Ogbomo, P. Ukueje, K. Ukaoha, and B. Malasowe, "Enhanced Model For Intrusion Detection In A Cloud Environment," The Journals of the Nigerian Association of Mathematical Physics, vol. 69, pp. 111-127, 2025.
3. S. Almabdy and A. Ullah, "Optimising Intrusion Detection Systems in Cloud-Edge Continuum with Knowledge Distillation for Privacy-Preserving and Efficient Communication," in 2025 IEEE 9th International Conference on Fog and Edge Computing (ICFEC), 2025, pp. 1-5.
4. H. Liu and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: A survey," applied sciences, vol. 9, p. 4396, 2019.
5. R. I. Farhan, A. T. Maolood, and N. F. Hassan, "Optimized deep learning with binary PSO for intrusion detection on CSE-CIC-IDS2018 dataset," Journal of Al-Qadisiyah for computer science and mathematics, vol. 12, pp. Page 16-27, 2020.
6. A. A. Hagar and B. W. Gawali, "Deep learning for improving attack detection system using CSE-CICIDS2018," NeuroQuantology, vol. 20, pp. 3064-3074, 2022.
7. M. A. Khan, "HCRNNIDS: Hybrid convolutional recurrent neural network-based network intrusion detection system," Processes, vol. 9, p. 834, 2021.
8. S. Alzughaibi and S. El Khediri, "A cloud intrusion detection systems based on dnn using backpropagation and pso on the cse-cicids2018 dataset," Applied Sciences, vol. 13, p. 2276, 2023.
9. I. U. Hewapathirana, "A comparative study of two-stage intrusion detection using modern machine learning approaches on the CSE-CICIDS2018 dataset," Knowledge, vol. 5, p. 6, 2025.

10. H. Dhillon, "Building effective network security frameworks using deep transfer learning techniques," The University of Western Ontario (Canada), 2021.
11. M. N. Ali, G. Tan, and A. Hussain, "Bidirectional recurrent neural network approach for Arabic named entity recognition," Future Internet, vol. 10, p. 123, 2018.
12. K. Sethi, Y. V. Madhav, R. Kumar, and P. Bera, "Attention based multi-agent intrusion detection systems using reinforcement learning," Journal of Information Security and Applications, vol. 61, p. 102923, 2021.
13. M. Bakro, R. R. Kumar, M. Husain, Z. Ashraf, A. Ali, S. I. Yaqoob, et al., "Building a cloud-IDS by hybrid bio-inspired feature selection algorithms along with random forest model," Ieee Access, vol. 12, pp. 8846-8874, 2024.
14. Aws (2018). A Realistic Cyber Defense Dataset (CSE-CIC-IDS2018):https://www.unb.ca/cic/datasets/ids2018.html Accseced: Jel.18,2025.
15. M. Bacevicius and A. PaulauskaiteTaraseviciene, "Machine learning algorithms for raw and unbalanced intrusion detection data in a multi-class classification problem," Applied Sciences, vol. 13, p. 7328, 2023.
16. S. Lee, D. Roh, J. Yu, D. Moon, J. Lee, and J.-H. Bae, "Deep feature fusion via transfer learning for multi-class network intrusion detection," Applied Sciences, vol. 15, p. 4851, 2025.
17. D. Moussaoui, M. Hadjila, W. Ferhi, and M. H. Hachemi, "Deep Learning-Based Anomaly and Intrusion Detection Using the CSE-CICIDS2018 Dataset," Engineering, Technology & Applied Science Research, vol. 15, pp. 2478224787, 2025.
18. A. Das, N. Shobha, M. Natesh, and G. Tiwary, "An Enhanced Hybrid Deep Learning Model to Enhance Network Intrusion Detection Capabilities for Cybersecurity," Journal of Machine and Computing, vol. 4, no. 2, p. 472, 2024.
19. A. F. Al-zubidi, A. K. Farhan, and S. M. Towfek, "Predicting DoS and DDoS attacks in network security scenarios using a hybrid deep learning model," Journal of Intelligent Systems, vol. 33, no. 1, p. 20230195, 2024.
20. S. Alshattnawi and H. R. Alshboul, "Combined Deep Learning Approaches for Intrusion Detection Systems," International Journal of Interactive Mobile Technologies, vol. 18, no. 19, 2024.
21. Y.-C. Wang, Y.-C. Houng, H.-X. Chen, and S.-M. Tseng, "Network anomaly intrusion detection based on deep learning approach," Sensors, vol. 23, no. 4, p. 2171, 2023.
22. S. M. Saleh, I. M. Sayem, N. Madhavji, and J. Steinbacher, "Advancing software security and reliability in cloud platforms through AIbased anomaly detection," in Proceedings of the 2024 on Cloud Computing Security Workshop, 2024, pp. 43-52.
23. M. N. Rahman and S. M. R. H. Nijhum, "Recurrent Neural Network Based Hybrid Deep Learning Architecture for Enhanced Network Intrusion Detection," in 2024 IEEE International Conference on Power, Electrical, Electronics and Industrial Applications (PEEIACON), 2024, pp. 400-405.