Journal of Engineering Managment and Soft Computing (JEMSC)

Identification and Evaluation of Deep Learning-Based Defense Strategies to Counter Deep Neural Network Attacks in Cloud Computing Environments

Ali Sayyah✉

Department of Computer Science, University of Tabriz, Tabriz, Iran, Email: ali.sayyah2@gmail.com

Article Info

Article type:
Research Article

Article history:
Received 23 October 2025
Received in revised form 4 December 2025
Accepted 28 December 2025
Published online 1 January 2026

Keywords:
defense strategies, deep learning, DDoS attacks, cloud computing, attack tools.

ABSTRACT

Distributed Denial of Service (DDoS) attacks are considered serious threats in the field of cloud computing and have always been a concern for system administrators. Identifying and detecting these attacks without knowing their tools and origins is a complex and challenging process. In the present study, the main goal is to provide deep learning-based defensive strategies to mitigate the effects of DDoS attacks in cloud computing environments. Based on previous studies, a model with five inputs including packet size, flow rate, flow duration, TCP flags, and byte volume, was extracted. For the implementation of the proposed model, deep learning algorithms, specifically Long Short-Term Memory (LSTM) neural network, Recurrent Neural Network (RNN), and Deep Neural Network (DNN), were used. Evaluation results showed that the LSTM algorithm has the best performance with 95% accuracy, followed by the RNN network with 93% and the DNN network with 92% accuracy. Additionally, among the input variables, flow rate had the most significant impact on attack detection, followed by packet size, byte volume, TCP flags, and flow duration in subsequent ranks of importance.

Cite this article: Sayyah, A, (2025)., Identification and Evaluation of Deep Learning-Based Defense Strategies to Counter Deep Neural Network Attacks in Cloud Computing Environments. *Journal of Engineering Management and Soft Computing*, 12 (1). 36-48.

DOI: https://doi.org/ 10.22091/jemsc.2026.14316.1317

DOI: https://doi.org/10.22091/jemsc.2026.14316.1317

Publisher: University of Qom

1) Introduction

Cloud computing forms the basis of modern digital infrastructure, offering more cost-effective and agile solutions for computing systems (Sarabadani et al., 2023). These services enable organizations to optimize operations and scale resources according to demand, making advanced computing accessible to a wide range of users (Kamalakkannan, 2025). The transition from traditional computing with its overhead and maintenance costs has transformed the landscape, allowing smaller enterprises to participate in technology-driven markets (Batchu et al., 2024). Despite all the advantages of cloud computing systems, they are not immune to risks and attacks. Cloud services are subject to DDoS attacks, which require intelligent security mechanisms for detection and response (Clinton et al., 2024). These types of attacks are considered a significant and major threat in computer security, causing service disruptions by saturating the network with high traffic, leading to system performance degradation and consequently making the network unavailable to its users (Wang et al., 2024). DDoS attacks occur through the use of botnets and a network full of infected devices that send a high volume of traffic to target systems to disrupt their accessibility (Abdullah & Bouke, 2024).

Cloud environments are vulnerable to these attacks. One reason is that DDoS attackers interfere with the reliability and continuity of service by using systems as reflectors or amplifiers (Alashhab et al., 2022; Hemmati et al., 2025). The main issue is detecting these attacks, which is naturally not an easy task and involves many complexities. On the other hand, understanding the origin and tools used in the attack, as well as the type of attack, can help formulate appropriate strategies to counter it. For example, attack tools can be categorized into user interface-based tools, attack rate dynamics-based tools, attack category-based tools, and attack target-based tools. Understanding these tools helps in formulating strategies to counter attacks (Ahirwal et al., 2025). Furthermore, by differentiating the type of attack, which can include volumetric attacks, protocol attacks, application-layer attacks, and other types of attacks, strategies can be developed based on the attack type.

Among these, deep learning algorithms, given their high capabilities in detecting other attacks in computing systems, can also be used to detect DDoS attacks. Although a large volume of research focuses on machine learning algorithms that have been successful in this area, deep learning algorithms have also been used in a few studies. The innovation of the present research lies in considering five variables: packet size, flow rate, TCP flags, bytes, and flow duration as input variables, and categorizing the output into two sections: attack tool and attack type. In fact, DDoS attacks in the present research are considered separately by attack tool and attack type, which has received less attention in previous research. Attack tools include interface-based tools, attack rate dynamics, attack category, and attack target; the attack type is also categorized into four types: volumetric attacks, protocol attacks, application-layer attacks, and other types of attacks. This output categorization and the type of inputs to the deep learning model form the innovative aspect of the present research.

Therefore, the goal of the present research is to formulate strategies to counter DDoS attacks using deep learning algorithms, which is done based on the type of attack and the tool used by the attacker. Finally, by implementing the above method, the researcher seeks to address the question: How is the identification of deep learning-based defense strategies to reduce DDoS attacks in cloud computing environments based on DDoS tools and attack types?

The structure of the present article is as follows: In the next section, the research gap is extracted and a literature review is presented. Subsequently, the methodology is presented, and then, the analysis is performed based on the methodology. Finally, the conclusion is described.

2) Research Background

The literature review in this section focuses solely on DDoS attacks, as this is the focus and scope of the current research. Therefore, the latest research in this area is reviewed, and finally, a research gap is extracted based on the conducted research. The extracted articles are primarily implemented methodologically using deep learning and machine learning algorithms. Gupta et al. (2021) propose a

big data and deep learning-based approach for distributed denial-of-service detection in cloud computing environments. Alashhab et al. (2022) address distributed denial-of-service attacks in cloud computing environments, discussing challenges, issues, and classification in this regard. Sanjalawe and Althobaiti (2023) focus on detecting distributed denial-of-service attacks in cloud computing based on associative feature selection and deep learning. Balasubramaniam et al. (2023) address the optimization of distributed denial-of-service detection based on deep learning in cloud computing.

Aljuaid and Alshamrani (2024) use a deep learning approach for intrusion detection systems in cloud computing environments. Reddy et al. (2024) implement machine learning techniques for cloud security in detecting distributed denial-of-service attacks. Batchu et al. (2024) propose a novel optimization-based deep learning framework for detecting distributed denial-of-service attacks. Clinton et al. (2024) address the classification of distributed denial-of-service attack traffic in SDN network environments using deep learning. Abdullah and Bouke (2024) address image-based network traffic pattern detection for distributed denial-of-service attacks in cloud computing environments. Wang et al. (2024) address predictive optimization of distributed denial-of-service attack mitigation in distributed systems using machine learning.

Afraji et al. (2025) highlight deep learning-based defense strategies for mitigating DDoS attacks in cloud computing environments. Ahirwal et al. (2025) address distributed denial-of-service attacks in cloud computing based on deep learning. Alhammadi and Mabrouk (2025) propose a multi-agent deep learning model for protecting cloud computing environments against distributed denial-of-service attacks. Alazmi and Alharbi (2025) investigate machine learning-based classification for denial-of-service attack detection in cloud computing. Kamalakkannan (2025) propose a deep learning model with optimization strategies for DDoS attack detection in cloud computing. Berríos et al. (2025) use a machine learning-based approach for detecting and mitigating distributed denial-of-service attacks in IoT environments.

Based on the literature review, it can be observed that most research focuses on DDoS attacks, and some of them also use deep learning algorithms. However, among the above research, the formulation of a strategy based on the type of attack and the attacker's tool for DDoS attacks is not observed, and thus, it can be said that there is a research gap in this area. The current research attempts to address this gap by proposing a new model in the field of DDoS attack detection and formulating a strategy to counter these attacks in cloud computing environments.

3) Research Method

The current research is applied and developmental in nature, utilizing deep learning algorithms. The algorithms used include LSTM neural networks, RNNs, and DNNs. The input variables for implementing the model include packet size, flow rate, TCP flags, bytes, flow duration, and protocols, which can determine a DDoS attack. These characteristics are recurring features in various DDoS attack detection datasets, the most important of which include the following datasets:

- UNSW-NB15
- NSL-KDD
- CICIDS2017
- CAIDA
- BOT_IOT

Based on the combination of the aforementioned datasets, the deep learning model presented in this research, which encompasses common DDoS attacks in cloud computing environments, is implemented. There is no specific ratio of tools or attack types; rather, there is a combination of various attack types and tools in the mentioned dataset. By training the data, they can determine which tool and which type of attack each input data represents. Therefore, the resulting output is based on two types of output:

- DDoS attack tool
- DDoS attack type

The tools and attack types are further introduced in Table 1.

Table 1) Classification of DDoS Attack Tools

Row	Medium	Dynamics of Attack Rate	Attack Category	Attack Target
1	Command Line Interface	Continuous	Resource Reduction	Link
2	Graphical Line Interface	Variable	Bandwidth Reduction	Final Point
3			Both Modes	

The attack tools are introduced in the table above and categorized based on each category, which includes the following four main categories:

1 . Interface

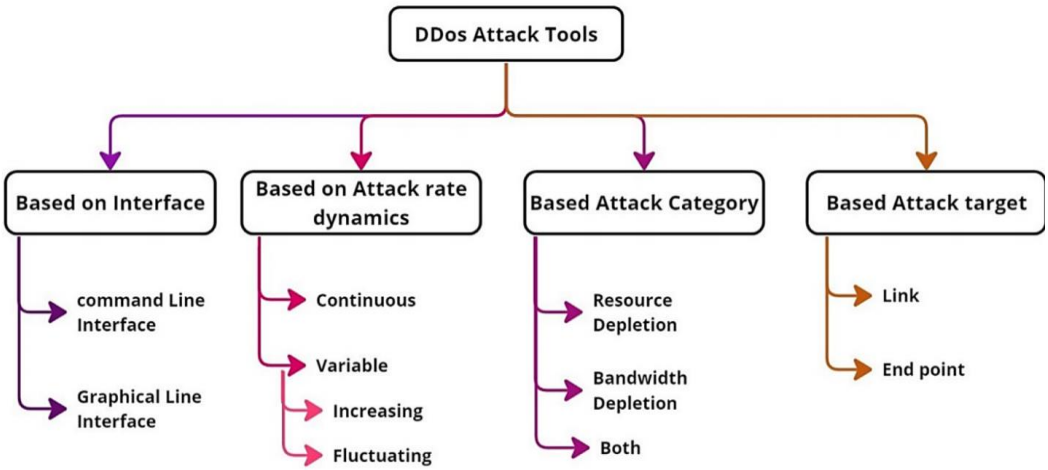
2 . Attack rate dynamism

3 . Attack category

4 . Attack target

The classification of various attack tools is presented in Figure 1.

Figure 1) Classification of DDoS Attack Types by Attacker Tools



The output categories examined in the present study include various attack types, which are divided into four general categories:

- Volumetric attacks
- Protocol attacks
- Application-layer attacks
- Other attacks

Finally, the model of the present study can be observed in Table 2 and Figure 2 below.

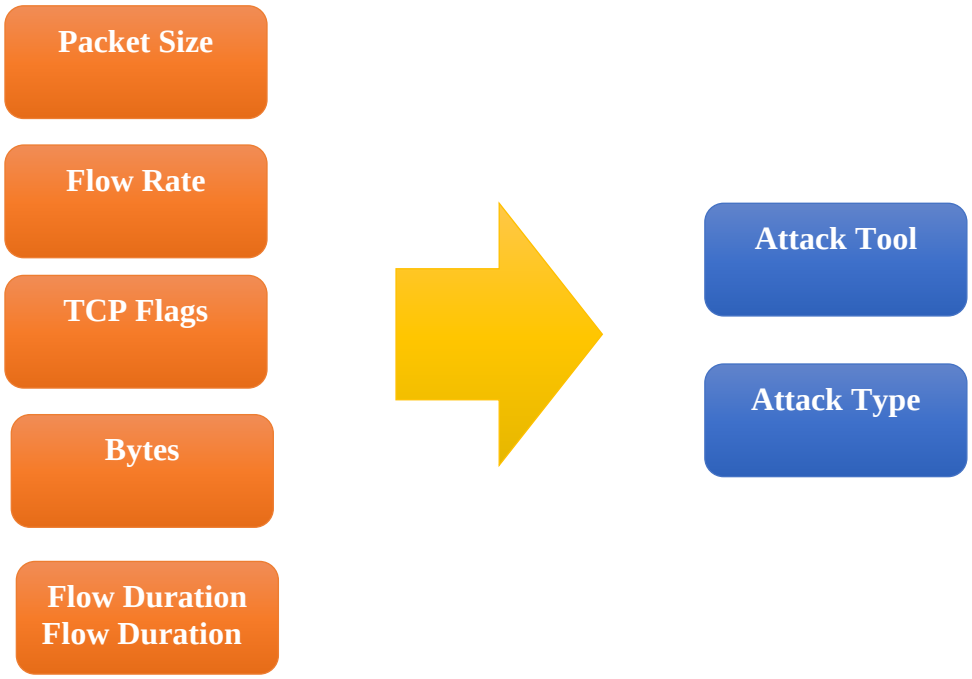
Table 2) Final Model of the Present Study

Row	Variable title	Variable symbol	Variable type	Variable scale
1	Packet size	X1	Input	Slight
2	Flow rate	X2	Input	Slight
3	TCP flags	X3	Input	Slight

Row	Variable title	Variable symbol	Variable type	Variable scale
4	Bytes	X4	Input	Slight
5	Flow duration	X5	Input	Slight
6	Attack tool	X6	Output	Nominal
7	Attack type	X7	Output	Nominal

Based on Table 2, it can be observed that the output is of a multi-class type.

Figure 2) The Final Model of the Present Study



Therefore, the deep learning algorithms implemented in the present study aim to classify attacks on tools as well as types of attacks, a classification that has not been performed in previous research using deep learning algorithms. As mentioned, the goal of the presented deep learning algorithms is to classify attacks. The classification criteria are measured based on the following four indicators:

- Accuracy
- Precision
- F1score
- Recall

The method of calculating the above criteria will be explained below. The higher the score obtained from the algorithms for these criteria, the higher the efficiency of the corresponding algorithm. Accuracy indicates the number of correctly classified samples relative to the total sample data. Its calculation formula is as follows:

$$accuracy = \frac{TN + TP}{TN + FP + TP + FN} \quad (1)$$

In the above relationship:

TN is the total number of true negatives.

TP is the total number of true positives.

FP is the total number of false positives.

FN is the total number of false negatives.

Precision indicates the positive predictive value in classifying data samples. Its formula is as follows:

$$Precision = \frac{TP}{FP + TP} \quad (2)$$

The next metric is recall, which is defined as sensitivity or true positive rate. Its formula is as follows:

$$Recall = \frac{TP}{FN + TP} \quad (3)$$

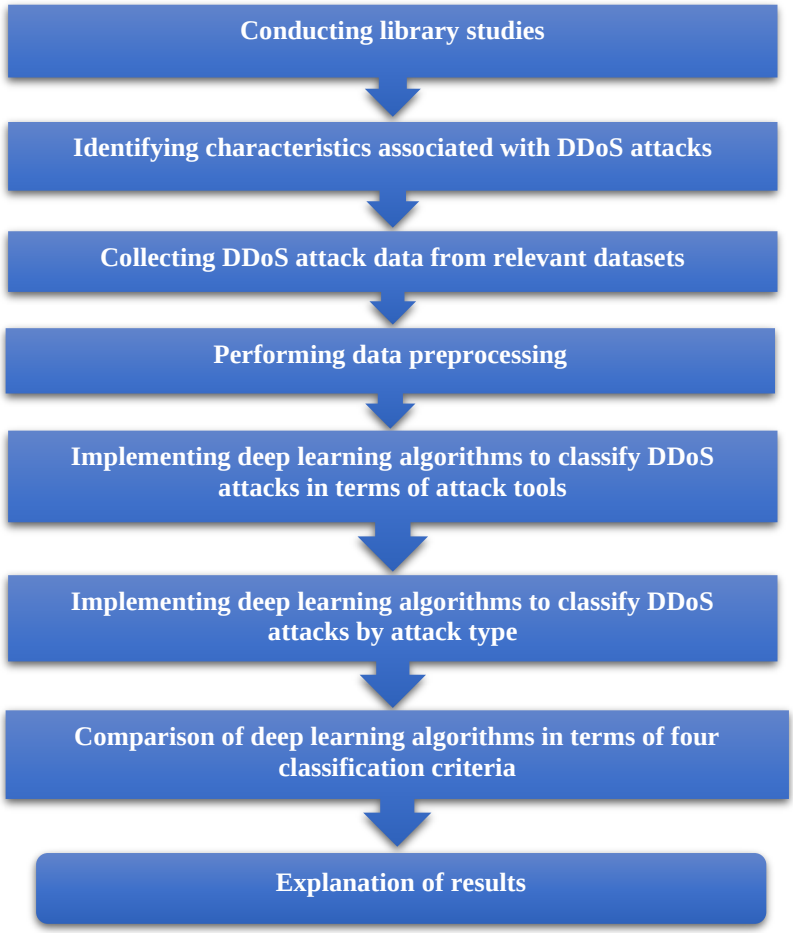
And finally, the ultimate criterion for evaluating the efficiency of machine learning algorithms in classification is the F1Score, which simultaneously calculates both precision and recall, and is as follows.

$$F1\ score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (4)$$

In the above formula, precision is multiplied by recall in the numerator, while in the denominator, these two metrics are added together and multiplied by 2, the result of which is the f1score value, and the higher it is, the better the performance of an algorithm.

The steps for conducting the research are presented in the flowchart below:

Figure 3) Research Steps



The following sections introduce data preprocessing, hyperparameter tuning, determining the number of neurons, and validating the deep learning algorithms used in this research. Data cleaning is a vital process in data analysis and is widely used in various data-related fields. Below, we examine six basic steps for data cleaning:

Data Quality Assessment: Data cleaning begins with a thorough examination of existing data to identify problems and weaknesses, including identifying relationships between data and assessing their diversity and quality.

Removing Duplicate or Inappropriate Items: Through duplicate removal techniques, redundant and irrelevant data are eliminated to increase the accuracy and efficiency of the dataset.

Correcting Structural Errors: In this step, structural errors such as different date, numerical, or unit of measurement formats in various columns are corrected.

Correcting Deviations: Unusual or incorrect values in the data are identified and removed from the dataset.

Checking for Missing Data: Missing or lost data are identified, and methods for managing them are applied to reduce their negative impact on analyses.

Validating Cleaned Data: Finally, the cleaned dataset is evaluated and compared with a reference database to ensure that the data have been properly cleaned.

Removing undesirable data is the first task in data cleaning. Removing undesirable samples refers to cleaning duplicate, redundant, or irrelevant data from the dataset. Then, missing data must be managed. Missing data is one of the common problems in datasets, arising from human error, system error, or challenges in data collection. Techniques such as "imputation" and "deletion" are used to solve the problem of missing data.

Another type of data, called "outlier" data, exists, which differs significantly from other samples. The presence of outliers affects the performance of machine learning models, and techniques such as "clustering," "interpolation," or "transformation" are used to manage them. Generally, "box plots," also known as "box-and-whisker plots," are used to examine outliers.

The next task in this regard is changing the data type. The process of changing the data type into a format that can be analyzed is called "data transformation." Data transformation uses methods such as "normalization," "scaling," and "encoding." The data transformation process consists of two parts: "data validation" and "data format change."

To normalize the data in this research, the scikit-learn library in Python is used. The Scikit-learn library provides a comprehensive set of tools for processes such as data preprocessing, "feature selection," "dimensionality reduction," building and training models, model evaluation, "hyperparameter tuning," and model sequencing.

In the next step, using the Min-Max normalization technique, we change the range of data values to the interval 0 to 1. To implement Min-Max normalization, we use the MinMaxScaler class from the Scikit-learn library.

The number of neurons in the input layer is equal to the number of data features. In very rare cases, there will be an input layer for bias. While the number of neurons in the output depends on whether the model is used as a regression or a classifier. If the model is a regression, the output layer has only one neuron. However, if the model is a classifier, depending on the model's class label, it will have one or multiple neurons. In the current research, given five input variables, there are five neurons in the input layer; nevertheless, since the model is a classification type, there is no one neuron in the output layer, and due to the presence of four labels for the model class, there are four neurons in the output layer.

Regarding hidden layers, it should be emphasized that there are several methods for determining the correct number of neurons to use in hidden layers, including:

Between the size of the input layer and the size of the output layer.

2/3 of the input layer size plus the output layer size.

Less than twice the size of the input layer.

In this research, the third method is used, and less than twice the size of the input layer, which includes five neurons, is used for the hidden layers. Therefore, nine neurons are considered in the hidden layer.

Activation functions in neural networks determine whether a node should be active or inactive. In other words, these functions use simple mathematical calculations to determine whether the node's input is important to the network or should be ignored.

The role of the activation function in neural networks is to produce an output value using the node's input values. More specifically, the activation function maps the weighted sum of the node's input to values between 0 and 1 or -1 and 1 (depending on the activation function). Then, this function passes its final value to the next layer. For this reason, this function is also called a transfer function.

In the current research, the sigmoid function is used. This nonlinear activation function converts its input to a value in the range of 0 to 1. The larger the input value, the closer the output value of this function gets to 1. However, if the input value of this function is very small (a negative number), the output value of the sigmoid function gets closer to zero. The sigmoid function is considered a "monotonic" function, but its derivative is not a monotonic function.

The following table summarizes the number of neurons and the activation function.

Table 3) Determination of the Number of Input and Hidden Neurons and the Activation Function

Parameter	amount
Number of neurons in the input layer	5
Number of neurons in the hidden layer	9
Method of determining hidden layers	Less than twice the size of the input layer
Activation function	Sigmoid function
Activation function range	0 to 1

In deep learning, hyperparameters include variables used to tune a neural network, such as regularization and learning rate. The Python Scikit-Learn library, or similar other software, provides default hyperparameters for each model, but these values are usually not optimal for our specific problem. Determining the best hyperparameters is often impossible, but suitable values can be found through trial and error.

The best way to narrow down hyperparameter values for a problem is to test and evaluate a large number of values for each hyperparameter. Using the RandomizedSearchCV method from the Python Scikit-Learn library, a grid of hyperparameter value ranges can be defined, and samples of these values can be randomly selected and evaluated.

To validate algorithms, k-fold cross-validation is used. In this method, the dataset is divided into k different forms, each called a fold. The model is trained k times on these k folds. In this way, k accuracies are obtained, and finally, the average of these accuracies is calculated.

The data division method, separating training, test, and validation data, is as follows in the table below.

Table 4) Data Division by Training, Test, and Validation Data

Data Type	**Allocated percentage**
Training	70%
Testing	15%
Validation	15%

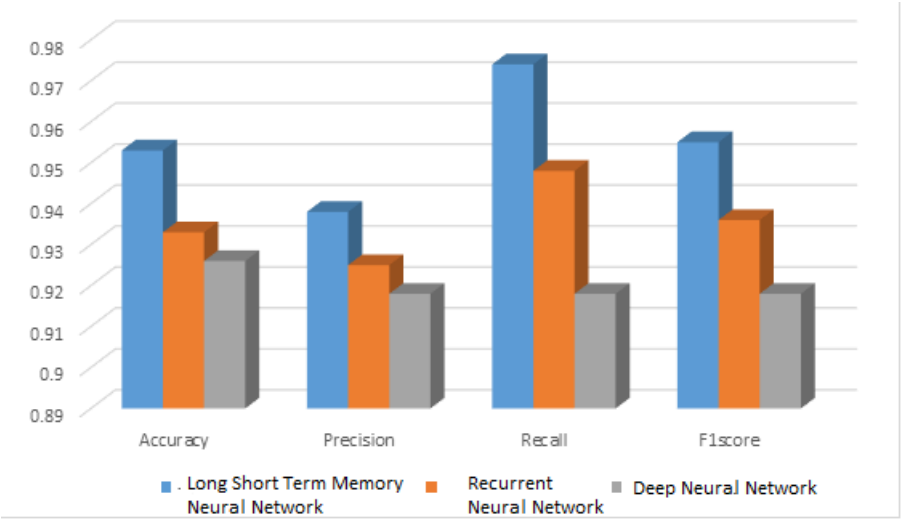
4) Research Findings

The analysis of the findings is presented below. First, the three algorithms introduced in the methodology section are implemented, and the algorithms are compared based on four metrics: accuracy, precision, recall, and f1score. The results are presented in Table 5.

Table 5) Comparison of Deep Learning Algorithms in DDoS Attack Detection

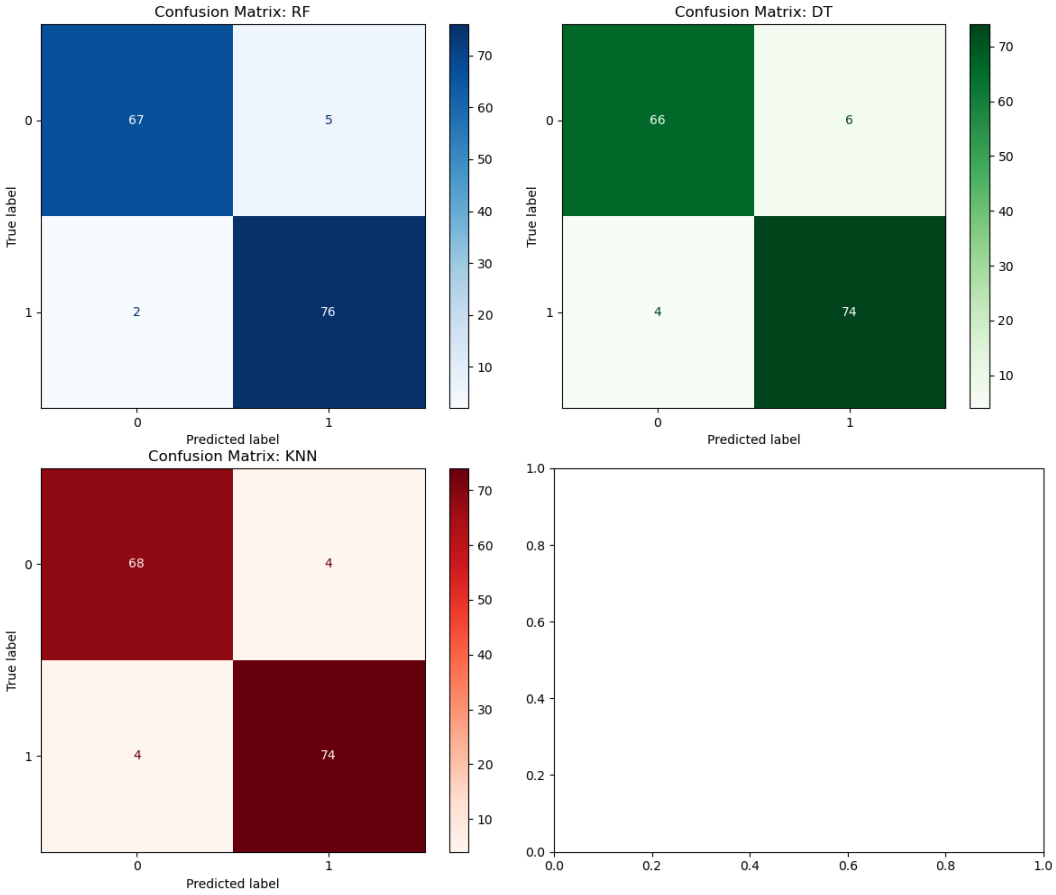
Algorithms	Accuracy	Precision	Recall	F1score
Long Short-Term Memory Neural Network	0.953	0.938	0.974	0.955
Recurrent Neural Network	0.933	0.925	0.948	0.936
Deep Neural Network	0.926	0.918	0.918	0.918

Figure 4) Comparison of Deep Learning Algorithms in DDoS Attack Detection



As can be seen, the LSTM neural network algorithm has the highest rate for all attack classification metrics, and therefore, can be considered the superior algorithm in the current research. Especially in terms of accuracy, which is the most important metric, it has a significant lead over other algorithms. Subsequently, the RNN algorithm is in second place in terms of importance and has achieved the second rank for all four metrics. The third rank belongs to the DNN algorithm, which indicates the weakest classification for this algorithm. The results will be further examined using the confusion matrix.

Figure 5) Confusion Matrix



Regarding the confusion matrix, it should be emphasized that a higher value on the main diagonal indicates better algorithm performance, and a lower value on the off-diagonal also confirms this performance. Based on Figure 5, it can be seen that the LSTM neural network algorithm is still the superior algorithm according to the confusion matrix, and thus, its suitable performance can be confirmed. Following this algorithm, are the RNN and the DNN, respectively.

Figure 6) ROC Curve for the Three Deep Learning Algorithms Used

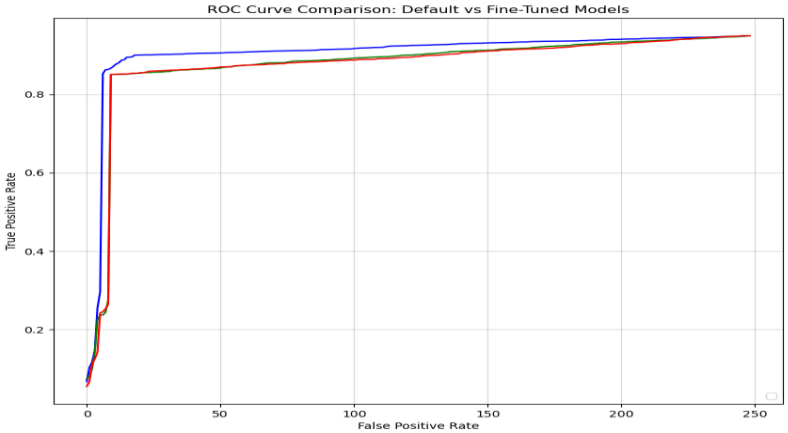


Figure 6 plots the ROC curve, which indicates the trend of achieving proper classification by each algorithm. The blue curve shows the performance of the LSTM neural network algorithm, while the red and green curves, which are very close to each other, show the performance of the RNN and DNN algorithms. By looking at the above graph, it can be observed that the LSTM neural network algorithm reveals better performance in terms of accuracy.

After identifying the superior algorithm in the present study, the next step is to examine the accuracy value for each output separately. In other words, it can be investigated that, considering only one of the tool outputs or the type of DDoS attack, to what extent the prediction or classification accuracy exists. The results are presented in Table 6.

Table 6) Investigating the Accuracy of Each Deep Learning Algorithm in Output Estimation

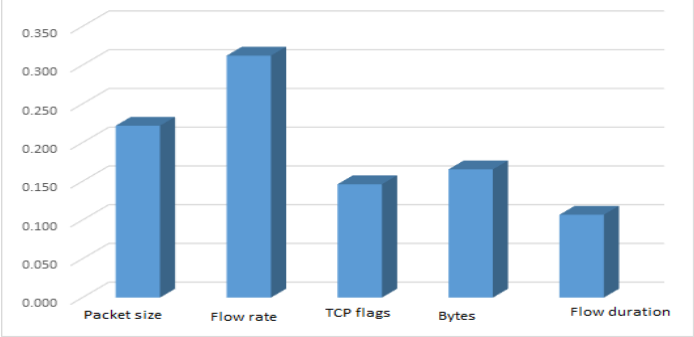
Output Categories	Long Short-Term Memory Neural Network	Recurrent Neural Network	Deep Neural Network
Interface	0.958506	0.934967	0.924008
Dynamics of attack rate	0.949234	0.936217	0.9159
Attack category	0.945648	0.921925	0.916933
Attack target	0.956945	0.926314	0.924658
Volume attacks	0.954242	0.92662	0.925815
Protocol attacks	0.941995	0.920771	0.914921
Software attacks	0.953741	0.920996	0.91033
Other attacks	0.949079	0.937502	0.920937

In Table 6, we observe relatively similar results for each output, indicating no significant difference in prediction. However, the algorithms achieved different accuracy values in their predictions, with the LSTM neural network naturally outperforming other algorithms. Subsequently, the impact of each input variable on the results, which essentially represents their contribution to the obtained accuracy or R, is presented in Table 7. It is worth noting that the Permutation Importance method in Python was used to assess the impact of each variable.

Table 7) Determining the Impact of Each Variable on Results

Row	Variable Name	Impact Percentage
1	Packet size	0.222
2	Flow rate	0.312
3	TCP flags	0.146
4	Bytes	0.166
5	Flow duration	0.107

Figure 7) Determining the Impact of Each Variable on the Results



Based on the results presented in Figure 7, it can be said that the flow rate has the most significant impact. Therefore, if we were to rank the five input variables, the flow rate would be assigned the highest importance, followed by packet size, bytes, TCP flags, and flow duration. Of course, it should be noted that the ranking was performed considering the existing and identified variables from the datasets introduced in the methodology section, and these results cannot be generalized to other datasets. However, overall, as the input variables have the highest frequency, it can be said that the results are largely verifiable.

5) Conclusion and Suggestions

DDoS attacks are a significant type of attack on cloud computing systems, making their detection very important. However, the crucial point in this regard is to formulate a strategy to counter these types of attacks. To understand the strategies for countering DDoS attacks, two criteria must be identified: the first criterion is the attack tool, and the second criterion is the type of attack, which was identified in the present study. Therefore, strategies for countering DDoS attacks can be considered tool-based and type-based strategies. Tool-based strategies must accurately identify the interface, consider the dynamics of the attack rate, and recognize the attack target, while type-based attack strategies must identify various types of attacks, including protocol, volumetric, application-layer, and other types of attacks, and have the necessary measures to counter these attack types. If there are false positives or false negatives in determining the strategy, it is expected that the counter-strategy may be misdiagnosed. This is because if the type of attack or the attack tool is misdiagnosed, the strategy will also be incorrectly determined, leading to an incorrect response to DDoS attacks.

In this study, machine learning algorithms were used to determine the attack tool and attack type, and it was found that five variables—flow rate, flow duration, packet size, TCP flags, and bytes—can lead to attack detection up to 95% and are influential variables in this regard. However, the ranking results of these features show that flow rate and packet size have the most significant impact, followed by bytes, TCP flags, and flow duration. The deep learning algorithms, used in the present study, were largely capable of performing the classification for the DDoS attack tool and type. Of course, the LSTM neural network algorithm outperformed other two algorithms, namely RNN and DNN. This is because this algorithm had higher values for all four metrics: accuracy, precision, f1-score, and recall. The LSTM neural network algorithm has consistently demonstrated its superior performance as a leading algorithm in similar problems, and therefore, the results obtained from the implementation of this algorithm are consistent with research conducted in this field.

It seems that the present study, based on attack tools and types, can provide an appropriate strategy, as the high volume of volumetric attacks requires a strategy to counter them, or protocol attacks require a strategy to manage them. On the other hand, application-layer attacks also play an important role among various types of attacks. Furthermore, issues such as user interface, attack rate dynamics, attack category, and attack target require strategies based on these tools. Consequently, it can be said that all these tools, according to the results obtained from the implementation of deep learning algorithms, require countering and management. Future research can extend the five-input model of the present study and investigate other variables and present their effect on the classification results.

The limitations of the present study are examined from three aspects, including limitations related to the dataset, model scalability, and the applicability and use of the model in the real world with high computational efficiency, which need to be addressed in future research.

References

Abdullah, A., & Bouke, M. A. (2024). Towards image-based network traffic pattern detection for DDoS attacks in cloud computing environments: A comparative study. In *CLOSER* (pp. 287-294).

Afraji, D. M. A. A., Lloret, J., & Peñalver, L. (2025). Deep learning-driven defense strategies for mitigating DDoS attacks in cloud computing environments. *Cyber Security and Applications*, 100085.

Ahirwal, M., Nema, P., & Richhariya, V. (2025). Distributed denial of service attacks in cloud computing based on deep learning: A study. *International Journal of Advanced Research and Multidisciplinary Trends (IJARMT)*, *2*(3), 28-40. https://ijarmt.com

Alashhab, Z. R., Anbar, M., Singh, M. M., Hasbullah, I. H., Jain, P., & Al-Amiedy, T. A. (2022). Distributed denial of service attacks against cloud computing environment: Survey, issues, challenges and coherent taxonomy. *Applied Sciences*, *12*(23), 12441. https://doi.org/10.3390/ app122312441

Alazmi, A. N. D., & Alharbi, Y. O. (2025, April). Classification-Based machine learning for detection of DDoS attack in cloud computing. In *2025 4th International Conference on Computing and Information Technology (ICCIT)* (pp. 210-214). IEEE.

Alhammadi, N. A. M., & Mabrouk, M. (2025). A multi-agent-based deep learning model for protecting cloud computing environment against distributed denial of service flooding attacks. *Journal of Soft Computing and Data Mining*, *6*(1), 406-422.

Aljuaid, W. A. H., & Alshamrani, S. S. (2024). A deep learning approach for intrusion detection systems in cloud computing environments. *Applied Sciences*, *14*(13), 5381. https://doi.org/10.3390/app14135381

Balasubramaniam, S., Vijesh Joe, C., Sivakumar, T. A., Prasanth, A., Satheesh Kumar, K., Kavitha, V., & Dhanaraj, R. K. (2023). Optimization enabled deep learning-based ddos attack detection in cloud computing. *International Journal of Intelligent Systems*, *2023*(1), 2039217. https://doi.org/10.1155/2023/2039217

Batchu, R. K., Bikku, T., Thota, S., Seetha, H., & Ayoade, A. A. (2024). A novel optimization-driven deep learning framework for the detection of DDoS attacks. *Scientific Reports*, *14*(1), 28024. https://doi.org/10.1038/s41598-024-77554-9

Berríos, S., Garcia, S., Hermosilla, P., & Allende-Cid, H. (2025). A machine-learning-based approach for the detection and mitigation of distributed denial-of-service attacks in Internet of Things environments. *Applied Sciences*, *15*(11), 6012. https:// doi.org/10.3390/app15116012

Clinton, U. B., Hoque, N., & Robindro Singh, K. (2024). Classification of DDoS attack traffic on SDN network environment using deep learning. *Cybersecurity*, *7*(1), 23. https://doi.org/10.1186/s42400-024-00219-7

Gupta, B. B., Gaurav, A., & Peraković, D. (2021, October). A big data and deep learning based approach for ddos detection in cloud computing environment. In *2021 IEEE 10th Global conference on consumer electronics (GCCE)* (pp. 287-290). IEEE.

Hemmati, A., Motevalli, S. H., Pourghader Chobar, A., Akhlaghpour, A., & Nazari, L. (2025). Analyzing customer sentiment with AI to improve the smart supply chain. *Engineering Management and Soft Computing*, *11*(1), 306-286. https://doi.org/10.22091/jemsc.2025.3654.1260

Kamalakkannan, S. (2025, February). Deep learning model with optimization strategies for DDoS attack detection in cloud computing. In *2025 3rd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)* (pp. 413-417). IEEE.

Reddy, P., Adetuwo, Y., & Jakkani, A. K. (2024). Implementation of machine learning techniques for cloud security in detection of ddos attacks. *International Journal of Computer Engineering and Technology (IJCET)*, *15*(2), 25-34.

Sanjalawe, Y., & Althobaiti, T. (2023). DDoS attack detection in cloud computing based on ensemble feature selection and deep learning. *Computers, Materials & Continua*, *75*(2). https://doi.org/ 10.32604/cmc.2023.037386

Sarabadani, A., Saffarie, M., & RahseparFard, K. (2023). A framework for automating e-government services based on artificial intelligence. *Engineering Management and Soft Computing*, *9*(2), 106-118. https://doi.org/ 10.22091/jemsc.2024.9256.1171

Wang, B., He, Y., Shui, Z., Xin, Q., & Lei, H. (2024). Predictive optimization of DDoS attack mitigation in distributed systems using machine learning. *Applied and Computational Engineering*, *64*(1), 89-94. https://doi.org/ 10.13140/RG.2.2.15938.39369