Contents lists available at *Dergipark*

Journal of Scientific Reports-A

journal homepage: *https://dergipark.org.tr/tr/pub/jsr-a*

E-ISSN: 2687-6167 ***Number 55, December 2023***

RESEARCH ARTICLE

Receive Date: 28.07.2023 *Accepted Date: 03.10.2023*

Deep learning-based distributed denial of service detection system in the cloud network

Emine Deniz[1], Soydan Serttaş[2,*]

[1]*Kütahya Dumlupınar University, Department of Computer Engineering, Kütahya, Türkiye, ORCID:0000-0003-0670-3578*
[2]*Kütahya Dumlupınar University, Department of Computer Engineering, Kütahya, Türkiye, ORCID:0000-00001-8887-8675*

Abstract

Cloud computing offers an efficient solution that enables businesses and users to deliver flexible and scalable services by sharing resources. However, this shared resource pool also exposes vulnerabilities to various cyber threats, such as Distributed Denial of Service (DDoS) attacks. These DDoS attacks, due to their potential impact, can be highly destructive and disruptive. They render servers unable to serve users, leading to system crashes. Moreover, they can severely tarnish the reputation of organizations and result in significant financial losses. Consequently, DDoS attacks are among the most critical threats faced by institutions and organizations.

The primary objective of this study is to identify and detect DDoS attacks within cloud computing environments. Given the challenges associated with acquiring a cloud-based dataset, the main motivation behind this research was to construct a dataset within a cloud-based system and subsequently evaluate the intrusion detection capabilities of deep learning (DL) algorithms using this dataset. Initially, an HTTP flood attack was executed after creating a network topology within the OpenStack framework. The study employed Convolutional Neural Network (CNN), Artificial Neural Network (ANN), and Long Short-Term Memory (LSTM) models for attack detection. The performance of these models was assessed using various measurement metrics, and it was found that the LSTM model delivered the most impressive results, achieving an accuracy rate of 98%.

Keywords: Distributed denial of service; Deep learning; Anomaly detection; Cybersecurity; Cloud computing; OpenStack; Convolutional neural network; Long short-term memory; Artificial neural networks.

Corresponding author.
e-mail: soydan.serttas@dpu.edu.tr

1. Introduction

In addition to the many benefits that the internet has brought to our daily lives, there is also the downside of increased vulnerability to cyber threats. Various attacks, each with its own level of intensity and risk, can inflict critical damage on users and organizations. Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks are among the most prevalent attacks in network communication. These attacks generate abnormal communication traffic on the network, causing the system to slow down compared to normal operations or rendering it unusable for regular users. DoS attacks disrupt a target system's ability to provide services, while DDoS attacks serve the same purpose but originate from multiple sources.

In 2017, the National Institute of Standards and Technology (NIST) detected DDoS attacks and unauthorized intrusions that resulted in losses of up to $65.5 billion [1]. According to research by Kaspersky, the average damage caused by a DDoS attack is $120,000 for small businesses and $2,000,000 for large enterprises [2]. Due to their high capacities and prolonged impact durations, DDoS attacks pose a significant threat to institutions and organizations [3]. In 2007, Estonia experienced a DDoS attack that affected numerous public institutions and individuals [4]. In October 2016, a DDoS attack on one of America's Domain Name System (DNS) service providers made headlines worldwide. Subsequently, in 2018, a massive DDoS attack targeted GitHub servers [5]. These incidents aren't limited to Europe and America; even a private Turkish bank server fell victim to a DDoS attack [6].

The integration of cloud computing with the internet and computer networks has exposed cloud services to cyber threats and attacks [7]. Cloud computing faces various security threats, with a significant concern being the threat to the availability of cloud services. Among these threats, the DDoS attack stands out, directly impacting the accessibility of cloud services. During a DDoS attack, malicious actors intentionally exhaust cloud resources, preventing legitimate users from accessing the cloud's services and resources.

These incidents underscore the need for a reliable method to detect DDoS attacks. Therefore, the development of an effective DDoS detection system plays a crucial role in monitoring suspicious activities in the cloud [8]. In recent years, researchers have turned to Machine Learning (ML) methods for DDoS attack detection. ML algorithms, primarily relying on datasets, can identify abnormal behaviors in the network [9]. However, ML algorithms based on datasets may be slow to detect new attacks. Consequently, DL has gained popularity [10].

The main contributions of this research are as follows:

- The study introduces a novel DDoS attack scenario designed within a general cloud environment.
- A new network topology was created using the OpenStack [11] software, a public cloud testing environment. Training and test datasets containing DDoS attacks were generated with the open-source OpenStack software, which facilitates the creation and management of virtual servers.
- Attack and normal data logs were recorded by executing unique DDoS attack scenarios on the created network.
- Feature extraction was conducted on the collected data
- TensorFlow [12] was integrated into the OpenStack software framework, enabling the development of ANN, CNN, and LSTM models within the cloud network environment.

The novelty of this DDoS detection system lies in its approach of simulating DDoS attacks on a cloud platform and utilizing deep learning algorithms for their detection.

2. Literature review

The University of Minnesota experienced a significant DDoS attack for the first time in 1999. The university's network was unavailable for two days [13]. Since that date, DDoS attacks have increasingly become the most prevalent type of attack on network communications. Fig. 1 illustrates that the number of DDoS attacks has been rising each year.

Over the past few years, numerous detection and prevention techniques have been reported to mitigate DDoS attacks [15-18]. Many studies utilize ML approaches, such as classification, clustering, and prediction methods [19].

Igbe proposed a Dendritic Cell Algorithm (DCA) that utilizes the concept of Artificial Immune System (AIS) for DDoS attack detection. The proposed algorithm achieved an accuracy of 96% in detecting attacks [20].

Elsayed et al. suggested a model that combines One-Class Support Vector Machine (OC-SVM) and LSTM models using an AutoEncoder (AE) to improve performance. However, the proposed model achieved 74% accuracy [21].

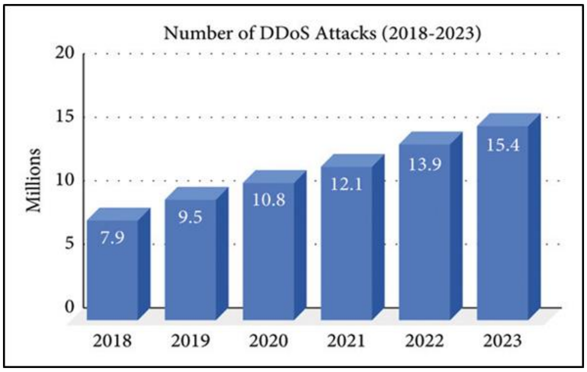

Fig. 1. Cisco's DDoS total attack volume history and forecast (2018-2023) [14].

Another study introduced a DL framework for network attack detection using LSTM and AE. The researchers conducted a comparison between the DL model and traditional ML algorithms using the NSL-KDD dataset. They showed that DL models outperformed ML models [22].

Wang et al. presented a novel model that combines AE with the SVM algorithm. After training, the SVM was utilized to detect anomalies within the extracted features. The model obtained an accuracy value of 88.73% in binary classification tasks using the NSL-KDD dataset [23].

Yavuz and Aygun proposed AE and Denoising AutoEncoder (DAE) techniques for attack detection. The DAE model provided with an accuracy of 88.65% [24].

Heikkonen and Farahnakian suggested an attack detection model that includes four AEs, where each AE output feeds into the next one in the current layer. The performance of the model was evaluated using the well-known KDD-CUP-99 dataset, achieving good performance with a detection rate of 94.53% [25].

Min et al. proposed a model using Memory-Augmented AutoEncoder (MemAE). The model achieved an F1-Score of 95% on the NSL-KDD dataset [26].

Anjum and Shreedhara introduced an ML method that relies on Semi-Supervised Learning for DDoS attack detection. According to their study, the proposed approach achieved an accuracy of 93% [27].

In the existing literature, there are also articles that analyze the utilization of ML in cloud computing [28]. Kushwah and Ali suggested an approach for detecting DDoS attacks in cloud environments consisting of Black Hole Optimization and ANN algorithms. The study used a dataset containing 12,500 training examples and 2,597 test examples. The highest accuracy achieved with the proposed method was reported to be 96% [29]. In another study, SVM and MLP classifiers were compared to detect DDoS TCP flood attacks in general clouds. The MLP model outperformed SVM with an accuracy of 94% compared to SVM's 92% [30]. Doshi et al. tested four different ML algorithms, SVM, DT, K-nearest neighbors (KNN), and Random Forest (RF), on normal and attack data collected from Internet of Things (IoT) based networks. The accuracy values of the four classifiers were 91%, 93%, 94%, and 99%, respectively [31].

Ma et al. proposed a multi-layered CNN model, showing that the CNN model outperformed SVM, DT, and Recurrent Neural Networks (RNN) in their study [32]. Potluri et al. also proposed a CNN-based detection approach, using a 3-layer and 2-dimensional (2D) CNN structure by converting data into images. The proposed CNN model achieved an accuracy of 91.14% in their study [33].

Ding and Zhai also proposed a CNN-based attack detection system. Their model consisted of a convolutional layer followed by three stacked stages with maximum pooling for feature extraction, forming a deep input features layer. The authors compared the outcomes of their model with traditional ML and DL methods using the NSL-KDD dataset. According to their findings, their proposed model demonstrated superior performance over the other approaches [34].

3. Materials and methodology

3.1. Cloud system

Cloud systems, commonly referred to as cloud computing, encompass the utilization of various services including servers, databases, data storage, networks, and software. These services are hosted within remote data centers and accessed via the Internet. Storing files in the cloud allows users to access their data from anywhere with an internet connection. Cloud computing is typically classified into two distinct approaches based on distribution models and service models, as illustrated in Fig. 2.

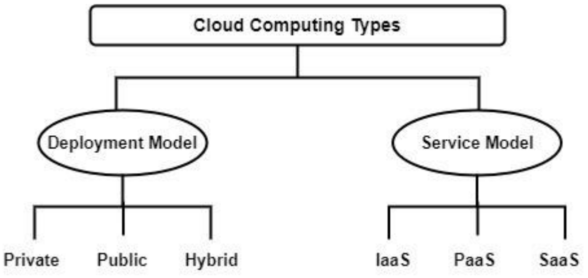

Fig. 2. Cloud computing types.

When considering the Distribution Model, it encompasses Public, Private, and Hybrid clouds. As for the Service Model, it includes IaaS (Infrastructure as a Service), PaaS (Platform as a Service), and SaaS (Software as a Service). Among these models, IaaS (Infrastructure as a Service) holds particular significance as it serves as the bedrock for flexible infrastructure resources, making it a fundamental component of cloud computing.

3.2. OpenStack

OpenStack is an open-source Infrastructure as a Service (IaaS) platform designed for hybrid and public cloud environments [11]. Its networking service ensures adaptable network connectivity in OpenStack cloud setups, maintaining a modular architecture to optimize resource utilization. Key components of OpenStack include:

- Nova: Manages virtual machines through APIs.
- Glance: Handles cloud image management and backups.
- Neutron: Manages virtual network structures.
- Cinder: Provides virtual block storage.

- Keystone: Ensures identity authentication and authorization.
- Swift: Offers additional storage services via HTTP API.

OpenStack is a modular, open-source cloud computing platform renowned for its scalability, support for hybrid clouds, and independence. Consequently, the network topology depicted in Fig. 3 was created to simulate a DDoS attack scenario within the OpenStack software environment.

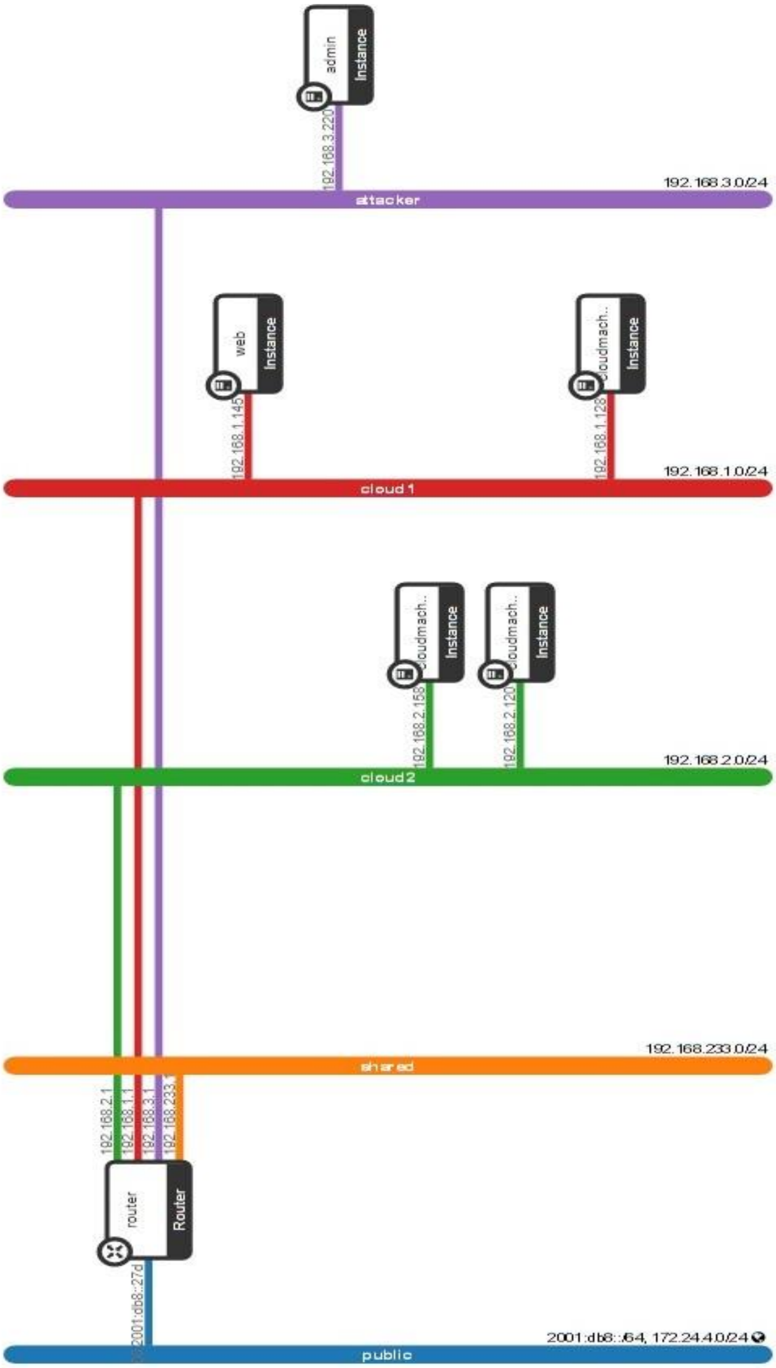

Fig. 3. Depicted network topology.

Each virtual instance is assigned an internal Internet Protocol (IP) address, used for communication among virtual instances belonging to the same tenants. In the context of the OpenStack cloud computing platform, a "tenant" is a structure used to isolate and manage resources like virtual machines, storage, and networks. For instance, various departments within an organization in the cloud communicate using these IP addresses. IP addresses starting with 192.x.x.x are considered internal. Tenants are separated from each other through distinct internal networks, as depicted in the OpenStack network topology shown in Figure 3. Tenant-1 is allocated to the orange network, tenant-2 operates within the green network, tenant-3 operates on the red network, and tenant-4 resides in the purple network. The virtual instance named 'web,' belonging to tenant-3, hosts a web server. All of these tenants establish external connectivity via a router connected to the public network. The web server on the red network, owned by tenant-3, was targeted in an HTTP flooding attack. Tenant-3 also has a virtual instance called 'cloudmachine-3,' which serves as a compute node. Virtual instances 'cloudmachine-1' and 'cloudmachine-2,' belonging to tenant-2, generated the HTTP flooding against the web server in the red network of tenant-3. Similarly, the administrator of tenant-4 initiated a flooding attack against the web server.

3.3. Dataset

A total of one hundred thousand rows of data were collected from the established network topology, comprising fifty thousand rows of normal data and fifty thousand rows of HTTP Flood attack data. Prior to utilizing the acquired data in standard ML and DL applications, preprocessing is necessary. The process of transforming raw data (in this study, pcap files) into structured formats to meet specific criteria is known as feature extraction. Each feature represents a dimension, variable, or quantity associated with a network traffic sample, which can be a packet, flow, or network during a defined time frame. Network traffic encompasses attributes such as source/destination IP addresses, protocol, transport ports, flags, and timestamps. Given that much of the information in pcap files might be unnecessary or not directly usable in DL applications, feature extraction was employed to prepare the data for further analysis.

3.4. Method

The flowchart depicting the process of this study is presented in Fig. 4. Initially, the network scenario described in section 3.2 of the OpenStack software was implemented to collect data related to both DDoS attacks and normal activity, employing the 'tcpdump' tool. Before this acquired data could be used in standard ML and DL applications, it had to undergo processing. The transformation of raw data into structured formats to meet specific criteria is known as feature extraction. The collected data was stored as a Pcap file, and using the Scapy library in Python, essential parameters were extracted from the raw data. These extracted features were then saved in a CSV file. Subsequently, these features were leveraged for attack detection through the use of DL models, such as ANN, CNN, and LSTM models.

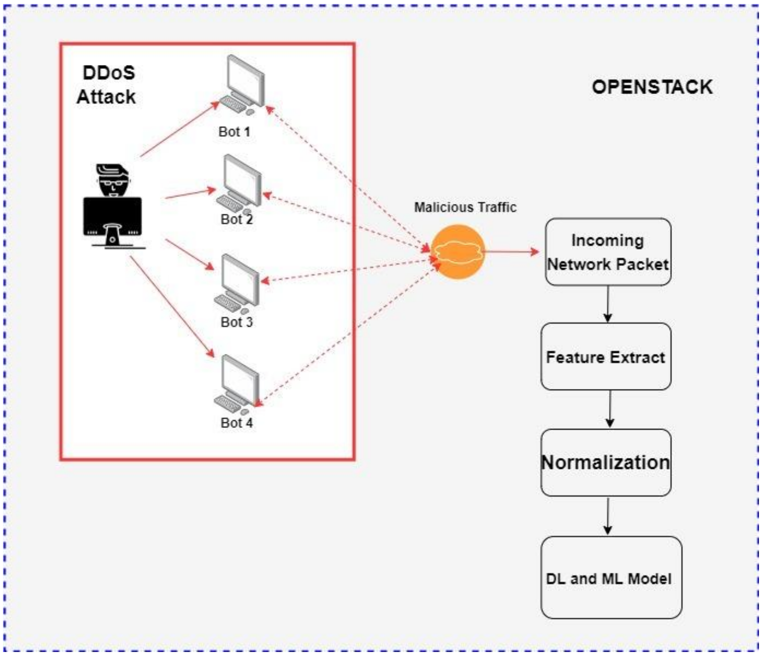

Fig. 4. The proposed system.

3.5. Evaluation

Performance metrics for evaluating deep learning methods are crucial. The typical performance evaluation criteria include accuracy, precision, F1-score, and recall. The ROC (Receiver Operating Characteristic) value is a metric employed to evaluate the performance of a classification model, especially in classification problems such as deep learning models. A higher ROC curve indicates better model performance.

The confusion matrix plays a vital role in providing a comprehensive assessment of the performance of a deep learning model and in understanding the nature of errors it makes. It aids in visualizing and quantifying the correlation between the actual values and the predictions generated by the model. The confusion matrix is constructed by comparing the actual outcomes with the model's predictions, allowing for the calculation of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) values.

Accuracy measures how well a prediction matches the actual outcome and is calculated using the formula shown in Equation 1.

$$Accuracy = \frac{TN+TP}{TN+TP+FP+FN} \tag{1}$$

Precision is a metric obtained by dividing the number of correctly detected DDoS attacks by the total number of detections labeled as true. It measures the system's accuracy in correctly identifying DDoS attacks, differentiating them from other attacks or normal flows. The formula for precision is defined in Equation 2.

$$Precision = \frac{TP}{FP+TP} \tag{2}$$

The recall is calculated by dividing the number of predicted attacks by the total number of actual attacks. As a result, it shows how many DDoS attacks were correctly identified and is also known as the "true positive rate." The formula for recall is provided in Equation 3.

$$Recall = \frac{TP}{FN+TP} \tag{3}$$

The F1 score is a metric that strikes a balance between precision and recall, producing a value between 0 and 1. The formula for F1-Score is defined in Equation 4.

$$F1\ Score = \frac{2*Precision*Recall}{Precision+Recall} \tag{4}$$

4. Results and discussion

4.1. DDoS attack detection using ANN model

An artificial neural network is a computational model inspired by the structure and functioning of the human brain, used for tasks such as pattern recognition and deep learning. Fig. 5 provides a detailed overview of the ANN model's architecture in this study. The initial dense layer employs the ReLU activation function, while the second layer uses the Sigmoid function. The model is compiled with the "CrossEntropy" loss function and employs the "Adam" optimizer.

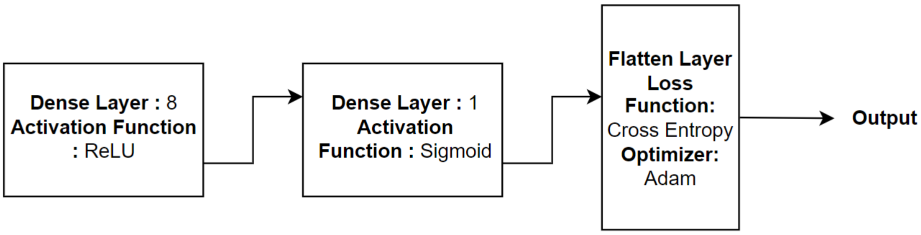
Dense Layer : 8
Activation Function : ReLU
Dense Layer : 1
Activation Function : Sigmoid
Flatten Layer
Loss Function: Cross Entropy
Optimizer: Adam
Output

Fig. 5. ANN model architecture.

The ANN model described here follows a two-layer architecture. Generally, the number of parameters within a specific layer of a model corresponds to the count of trainable elements within that layer. As shown in Table 1, the first layer comprises 208 trainable parameters, while the second layer consists of 9 trainable parameters. Consequently, the total count of trainable parameters in the ANN model amounts to 217.

Table 1. ANN model summary.

Layer	Output shape	#Parameters
Dense layer	(,8)	208
Dense layer	(,1)	9

After training the created ANN model with data obtained from the OpenStack platform, the accuracy value was obtained as 85%, the precision value was 98%, the recall value was 77%, and the F1 score was 86%. These evaluation metrics suggest the need for more advanced models in attack detection. In classification problems like this, metrics such as recall, precision, ROC value, and confusion matrix are examined, similar to deep learning models. Fig. 6 presents the ROC curve for the ANN model. When analyzing the ROC curve, it ideally should be close to 1. However, due to the limited sophistication of the ANN model, the desired ROC curve could not be achieved.

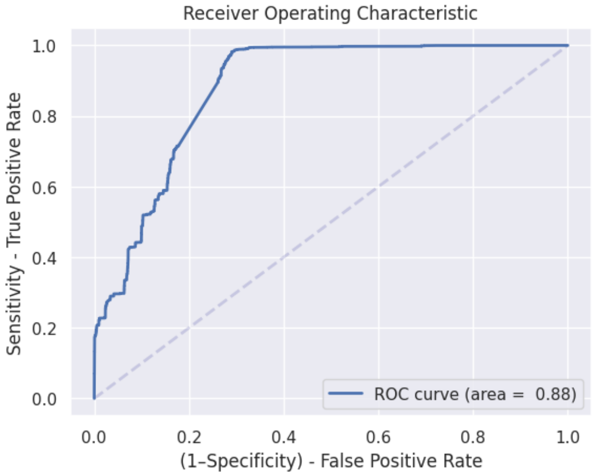

Fig. 6. ROC graph of ANN model.

The confusion matrix for the ANN model is also presented in Fig. 7. The ANN model successfully identified 12,752 instances of attack data but misclassified 489 attack instances as normal data. It correctly identified 16,945 out of 21,759 normal data instances as normal, but it erroneously labeled 4,814 normal data instances as attacks. The ANN model's classification of 4,814 normal data instances as attacks signifies a significant discrepancy. To address this issue and reduce this number, the development of more advanced models is warranted.

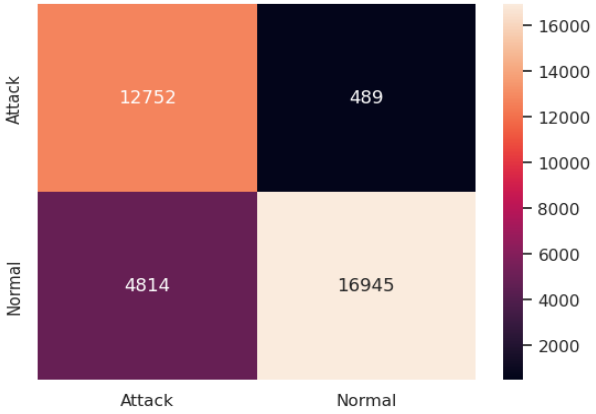

Fig. 7. Confusion Matrix of ANN model.

4.2. DDoS attack detection using CNN model

CNN, is a type of artificial neural network commonly applied in deep learning, particularly for tasks such as image processing and recognition. Fig. 8 shows a detailed overview of the CNN model architecture proposed by the authors of this study. The CNN model adopts a 10-layer architecture. A dropout layer is employed to mitigate overfitting in the CNN model, while a pooling layer is used to reduce data dimensions and optimize computations. The integration of dropout and pooling layers has been leveraged to enhance the performance and generalization capability of the CNN model utilized in this study. The "ReLU" activation function has applied across all layers. The stride value in CNN determines the steps at which data is shifted and influences the size of the feature map; a larger stride reduces both size and computation cost. Meanwhile, the padding value is utilized to either maintain or decrease the size of the feature map after convolution; 'same' padding retains the size, while 'valid' padding reduces it. In our study, 'same' padding has been employed to maintain the size of the feature map. To reduce the number of parameters and computation cost, a stride size of 2 has been selected. The model has compiled with the "CrossEntropy" loss function and the "Adam" optimizer.

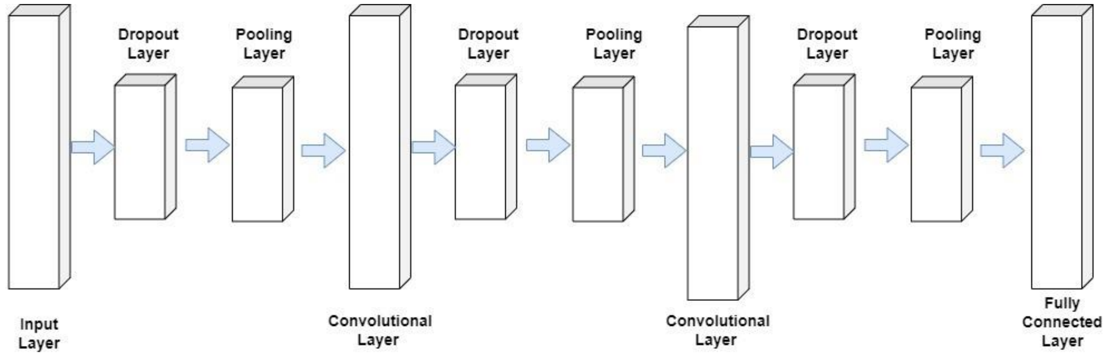

Fig. 8. Proposed CNN model architecture.

Convolutional layers are the layers trained in the CNN model while pooling layers have no trainable parameters. Table 2 displays the layers of the CNN model and the number of trainable parameters in each layer. The total number of trainable parameters in the CNN model is 22,561. The presence of a higher number of trainable parameters in the CNN model, as compared to the ANN model, has resulted in improved outcomes.

Table 2. CNN model summary.

Layer	Output shape	#Parameters
Input layer	(22,32)	1632
Dropout layer	(22,32)	0
Pooling layer	(11,32)	0
Convolutional layer	(11,64)	4160
Dropout layer	(11,64)	0
Pooling layer	(5,64)	0
Convolutional layer	(5,128)	16512
Dropout layer	(5,128)	0
Pooling layer	(2,128)	0
Fully connected layer	(,1)	257

After training the created CNN model with data obtained from the OpenStack platform, the accuracy value was obtained as 97%, the precision value was 98%, the recall value was 92%, and the F1-score was 95%. Figure 9 provides the ROC curve for the CNN model. The ROC curve of the CNN model is close to 1 indicating that the CNN model is more successful than the ANN model in the dataset obtained in this study.

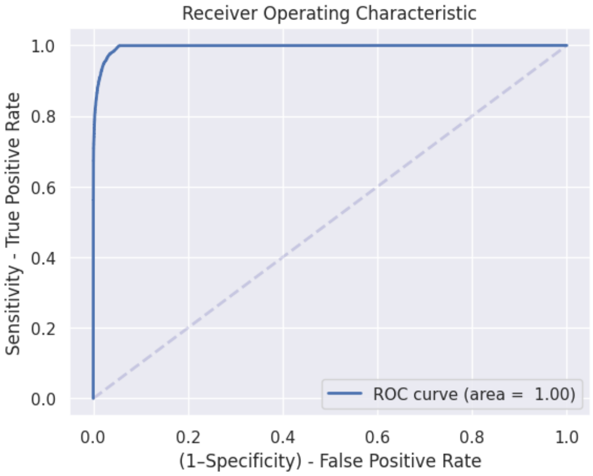

Fig. 9. ROC graph of CNN model.

The confusion matrix for the CNN model is shown in Figure 10. Due to their ability to automatically extract pertinent features, CNN models exhibit better performance compared to the ANN model. The CNN model correctly classified 32,365 instances of attack data as attacks but erroneously labeled 365 attack instances as normal data. It accurately identified 26,418 out of 29,053 normal data instances but misclassified 2,635 normal data instances as attacks. When analyzing the confusion matrix, it becomes apparent that the CNN model has higher accuracy in correctly identifying attack data as attacks and normal data as normal when compared to the ANN model.

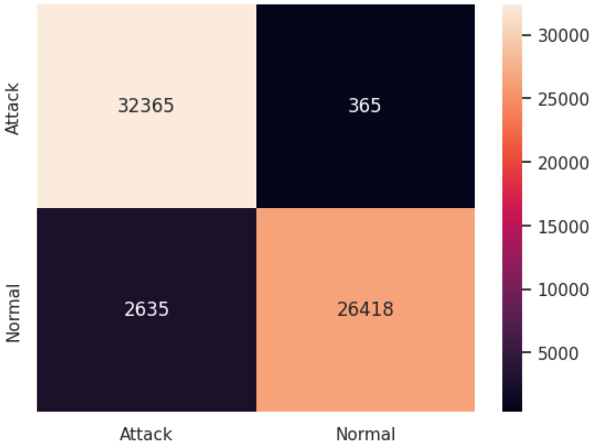

Fig. 10. Confusion Matrix of CNN model.

4.3. DDoS attack detection using LSTM model

LSTM is a specialized deep-learning architecture designed to capture and model long-term dependencies and sequential patterns in data. Figure 11 provides a detailed overview of the LSTM model architecture proposed by the authors of this study. The "ReLU" activation function is used in all layers. The model is compiled with the "CrossEntropy" loss function and the "Adam" optimizer.

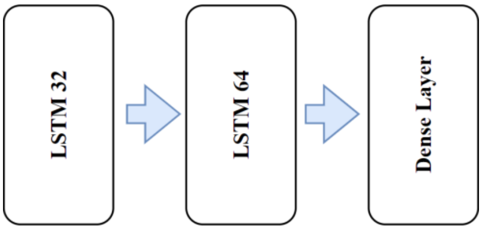

Fig. 11. LSTM model architecture.

The LSTM model has a 3-layer architecture. Table 3 shows that the first layer has 14,848 trainable parameters, the second layer has 4,160, and the last layer has 965. The total number of trainable parameters in the LSTM model is 19,073.

Table 3. LSTM model summary.

Layer	Output shape	#Parameters
LSTM32 layer	(,32)	14848
LSTM64 layer	(,64)	4160
Dense layer	(,1)	965

After training the created LSTM model with data obtained from the OpenStack platform, the accuracy value was obtained as 98%, the precision value was 99%, the recall value was 94%, and the F1-score was 97%. Figure 12 provides the ROC curve for the LSTM model. The ROC curve of the LSTM model being close to 1 indicates that the LSTM model is more successful than the ANN and the CNN models in the dataset obtained in this study.

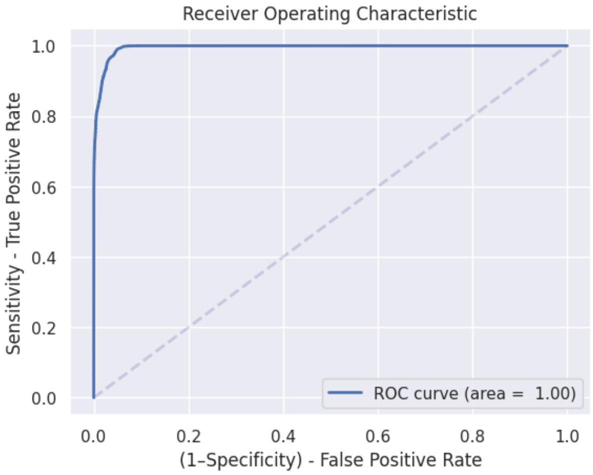

Fig. 12. ROC graph of LSTM model.

The confusion matrix of the LSTM model is presented in Figure 13. The LSTM model has correctly identified 33,234 attack instances as attacks, but misclassified 169 attack instances as normal data. It also correctly classified 34,727 normal instances as normal, but misclassified 1,854 normal instances as attacks. When examining the complexity matrix of the three models, it is evident that the LSTM model performed the best in correctly predicting attack instances as attacks and normal instances as normal.

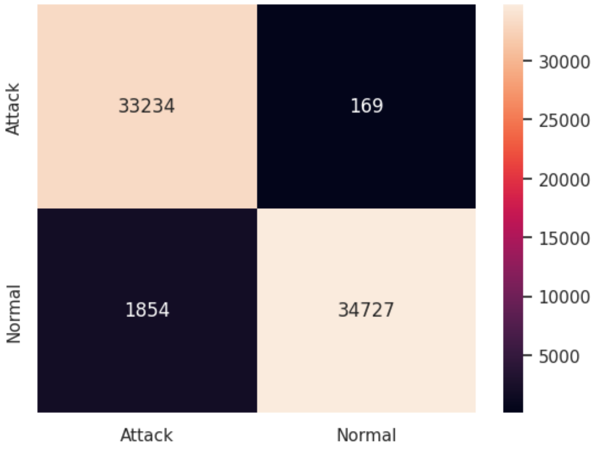

Fig. 13. Confusion Matrix of LSTM model.

4.4. Comparison of deep learning models

To determine which model excels in detecting DDoS attacks, we calculated accuracy, precision, recall, and F1-score values for the DL models. Table 4 presents the assessment metrics for these DL models. All DL models underwent training, validation, and testing using a 70:15:15 dataset split. The "Batch" size signifies the number of samples employed in each iteration during model training, typically ranging from 16 to 512. In this study, we set the

"Batch" sizes to 8, 16, 32, and 64, discovering that a "Batch" size of 32 yielded the most favorable results after DL model training. The "Epoch" number indicates how many times the entire training dataset is presented to the model during training. Each epoch encompasses a full iteration through the dataset, and conducting multiple epochs allows the model to learn from the data repeatedly, potentially enhancing its performance. In this study, we utilized 30 epochs. Furthermore, we implemented early stopping, a technique that interrupts the model training process when the validation loss starts to rise on the validation set. This approach aids in preventing overfitting and ensures that the model halts at an optimal point during training.

Table 4. Evaluation metrics and results of deep learning models (ANN, CNN, LSTM).

Model	Accuracy			Precision	Recall	F1 Score
	Train	Test	Validation			
ANN	0.85	0.85	0.84	0.98	0.77	0.86
CNN	0.97	0.97	0.97	0.98	0.92	0.95
LSTM	**0.98**	**0.98**	**0.98**	**0.99**	**0.94**	**0.97**

When analyzing Table 4, it becomes evident that the LSTM model boasts the highest accuracy at 98% on the acquired dataset. While accuracy is a valuable metric for assessing a model's performance, it should not stand alone. Precision values reveal that the LSTM model achieved 99%, while CNN and ANN both attained 98%. A closer examination of Recall and F1-Score metrics highlights that the LSTM model has significantly outperformed the ANN. The ROC AUC scores for these models were as follows: ANN achieved an AUC of 0.88, while CNN and LSTM both achieved perfect AUC scores of 1. The ROC AUC scores provide valuable insights into the discriminative power of these models. A perfect AUC score of 1 indicates that both CNN and LSTM achieved flawless discrimination between positive and negative classes. On the other hand, while ANN performed well with an AUC of 0.88, it exhibits slightly lower discriminative ability compared to CNN and LSTM.

4.5. Positioning the study in the literature

Cloud systems have increasingly become a crucial technology for data storage, processing, and analysis in today's world. However, as shown in Table 5, research in this field remains limited, highlighting the importance of this study in addressing the gaps in the limited literature related to cloud systems. One significant aspect of this research is the dataset obtained through the utilization of cloud systems. This innovative approach in data collection and processing has enabled more effective handling and analysis of large datasets, emphasizing the contributions and uniqueness of this study.

In contrast, while most of the studies listed in Table 5 rely on traditional machine learning methods such as decision trees, KNN, and SVM, this study adopts deep learning techniques, including LSTM and CNN. This study stands out by achieving exceptionally high accuracy rates compared to other similar research. Specifically, the model's extraordinary effectiveness is evident, with an exceptional 98% accuracy rate achieved through the use of LSTM. These results highlight the successful application of deep learning methods in this study and demonstrate important contributions to the field of data analysis.

Table 5. Linking study results to the literature.

References	Dataset	Methods	Accuracy	Area detection
Sofi, Mahajan, & Mansotra, 2017	public	DT, SVM, MLP	91%, 92%, 96%	-
Sharma, Mahajan, & Mansotra, 2016	public	DT, SVM, ANN	90%, 91%, 94%	-
Igbe, Ajayi, & Saadawi, 2017	public	DCA	96%	-

Elsayed M. S., Le-Khac, Dev, & Jurcut, 2020	public	AE with OC-SVM, LSTM	74%	-
Su, Sun, Zhu, Wang, & Li, 2020	public	AE ile SVM	88%	-
Aygun & Yavuz, 2017	public	AE, DAE	88%	-
Farahnakian & Heikkonen, 2018	public	AE	94%	-
Min, Yoo, Kim, Shin, & Shin, 2021	public	MemAE	95%	-
Derakhsh, Daneshjoo, & Delara, 2018	public	GA	64%	-
Anjum & S, 2019	public	Semi-Supervised Learning	93%	-
Kushwah & Ali, 2017	private	ANN	96%	yes
Sahi, Lai, Li, & Diykh, 2017	private	SVM, MLP	92%, 94%	yes
Doshi, Apthorpe, & Feamster, 2018	private	KNN, SVM, DT	91%, 93%, 94%	yes
Potluri, Ahmed, & Diedrich, 2018	public	CNN	91%	-
Zhang, Yu, & Li, 2018	public	XGBoost	89%	-
This study	**private**	**ANN, CNN, LSTM**	**85%, 97%, 98%**	**yes**

5. Conclusion

DDoS attacks, recognized as a major threat to networks and cloud systems, present substantial risks to computer system security due to their potential impact. ML and DL are two of the most widely explored technologies for intrusion detection systems. Nevertheless, the challenge of acquiring a cloud-based dataset for assessing ML and DL algorithms poses a significant hurdle. The absence of such a dataset served as the primary motivation for this research.

This study introduces the design of a cloud-based DDoS attack detection system and compares the performance of DL algorithms. We established a network topology using the OpenStack framework and gathered DDoS attack data through the simulation of HTTP flood attacks. Subsequently, we employed three distinct DL models - ANN, CNN, and LSTM algorithms - to detect attacks using the collected data. Our analysis encompassed multiple performance metrics, including accuracy, precision, recall, F1-score, and ROC curve assessments.

Among the DL models employed in this study, the LSTM model exhibited notably higher precision, recall, and F1-score values—99%, 94%, and 97%, respectively—compared to the ANN and CNN models, demonstrating its superior performance. The ROC curve analysis effectively illustrated how adeptly the models classified the labels. Based on their ROC curves, the CNN and LSTM models delivered the most promising results in this study.

For future research, we suggest diversifying the cloud network topology by creating various types of attack scenarios and expanding the dataset classes. Additionally, the concurrent execution of DL models and the incorporation of an ensemble voting technique could enhance attack detection capabilities.

Acknowledgements

There is no conflict of interest with any person/institution in the prepared article.

References

[1] M. Mittal, K. Kumar and S. Behal, “Deep learning approaches for detecting DDoS attacks: a systematic review”, Soft Computing, 1-37, 2022.

[2] D. Berard, “A single DDoS attack can cost a company more than $400,000”, https://www.kaspersky.com/about/press-releases/2015_a-single-ddos-attack-can-cost-a-company-more-than--400000, (accessed Jul. 27, 2023).

[3] C. Canongia, and R. A. Mandarino, “Cybersecurity: The new challenge of the information society”, In Handbook of Research on Business Social Networking: Organizational, Managerial, and Technological Dimensions, 165-184, 2012, doi:10.4018/978-1-4666-4707-7.ch003.

[4] A. D. Samsoerizal, E. R. Hidayat, and A. Sukendro, “Analytical study of indonesian cybersecurity: lesson learned from estonian cyberattacks in 2007”, International Journal of Arts and Social Science, 32-33, 2022.

[5] Balaban, “Denial-of-service attack”, Intel J. Info. Sec. and Cybercrime, 10-59, 2021.

[6] Rawashdeh, M. Alkasassbeh, & M. Al-Hawawreh, “An anomaly-based approach for DDoS attack detection in cloud environment”, International Journal of Computer Applications in Technology, 312-324, 2018.

[7] E. T. Ayan, M. S. Zengin, G. Deniz, H. A. Duru and B. Bardak, “Interpretable cybersecurity event detection in turkish: a novel dataset”, In 2022 Innovations in Intelligent Systems and Applications Conference, Antalya, Turkey, 2022, pp. 1-6, doi: 10.1109/ASYU56188.2022.9925501.

[8] R. V. Deshmukh, and K. K. Devadkar, “Understanding DDoS attack & its effect in cloud environment”, Procedia Computer Science, 202-210, 2015.

[9] N. Bindra, and M. Sood, “Detecting DDoS attacks using machine learning techniques and contemporary intrusion detection dataset”, Automatic Control and Computer Sciences, 419-428, 2019.

[10] X. Yuan, P. He, Q. Zhu, and X. Li, “Adversarial examples: attacks and defenses for deep learning”, IEEE transactions on neural networks and learning systems, 2805-2824, 2019.

[11] “The Most Widely Deployed Open Source Cloud Software in the World”, https://www.openstack.org/ (accessed Feb. 1, 2023).

[12] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, *et al.* “TensorFlow: large-scale machine learning on heterogeneous distributed systems”, http://download.tensorflow.org/paper/whitepaper2015.pdf, (accessed Jul. 20, 2023).

[13] G. C. Kessler, and D. E. Levin, “Denial-of-service attacks”, John Wiley and Sons, 12 September 2015, doi.org/10.1002/9781118851678.ch18.

[14] Cisco Annual Internet Report (2018–2023) White Paper, Available: https: //www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html, (accessed Jun. 6, 2020).

[15] Mirkovic, and P. Reiher, “A taxonomy of DDoS attack and DDoS defense mechanisms”, ACM SIGCOMM Comput. Commun. Rev., 34(2), 39-53, 2004.

[16] G. Carl, G. Kesidis, R. R. Brooks, and S. Rai, “Denial-of-service attack-detection techniques”, IEEE Internet computing, 10(1), 82-89, 2006.

[17] A. Y. Nur, and M. E. Tozal, “Record route IP traceback: combating DoS attacks and the variants”, Computers & Security, 72, 13-25, 2018.

[18] S. T. Zargar, J. Joshi, and D. Tipper, “A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks”, IEEE communications surveys & tutorials, 15(4), 2046-2069, 2013.

[19] R. Das, & T. H. Morris, “Machine learning and cyber security”, International Conference on Computer, Electrical and Communication Engineering - ICCECE, 2017, pp. 1-7.

[20] O. Igbe, O. Ajayi, and T. Saadawi, “Denial of service attack detection using dendritic cell algorithm”, 2017 IEEE 8th Annual Ubiquitous Computing, Electronics And Mobile Communication Conference, 2017, pp. 294-299.

[21] S. Elsayed, M. LE-Khac, N. A. Dev, and A. D. Jurcut, “Network anomaly detection using LSTM based

autoencoder", In Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks, 2020, pp. 37-45.

[22] M. S. Elsayed, N. A. Le-Khac, S. Dev, and A. D. Jurcut, "Ddosnet: A deep-learning model for detecting network attacks", A World of Wireless, Mobile and Multimedia Networks"(WoWMoM), 391-396, 2020.

[23] T. Su, H. Sun, J. Zhu, S. Wang, and Y. Li, "BAT: Deep learning methods on network intrusion detection using NSL-KDD dataset", IEEE Access, 8, 29575-29585, 2020.

[24] R. C. Aygun, and A. G. Yavuz, "Network anomaly detection with stochastically improved autoencoder-based models", In 2017 IEEE 4th International conference on cyber security and cloud computing (CSCloud), 2017, pp. 193-198.

[25] F. Farahnakian, and J. Heikkonen, "A deep auto-encoder based approach for intrusion detection system", In 2018 20th International Conference on Advanced Communication Technology (ICACT), 2018, pp. 178-183.

[26] Min, J. Yoo, S. Kim, and D. Shin, "Network anomaly detection using memory-augmented deep autoencoder", IEEE Access, 9, 104695-104706, 2021.

[27] M. Anjum, and K. S. Shreedhara, "Performance analysis of semi-supervised machine learning approach for DDoS detection", International Journal Of Innovative Research In Technology, 6(2), 144-147, 2019.

[28] Z. Zhong, M. Xu, M. A. Rodriguez, C. Xu, and R. Buyya, "Machine Learning-based Orchestration of Containers: A Taxonomy and Future Directions", ACM Comput. Surv. (CSUR), 2021.

[29] G. S. Kushwah, and S. T. Ali, "Detecting DDoS attacks in cloud computing using ANN and black hole optimization", 2nd International Conference on Telecommunication and Networks, pp. 1-5, 2017.

[30] A. Sahi, D. Lai, Y. Li, and M. Diykh, "An efficient DDoS TCP flood attack detection and prevention system in a cloud environment", IEEE Access, 5, 6036-6048, 2017, doi: 10.1109/ACCESS.2017.2688460.

[31] R. Doshi, N. Apthorpe, and N. Feamster, "Machine learning ddos detection for consumer internet of things devices" IEEE Security and Privacy Workshops (SPW), 2018, pp. 29-35.

[32] Ma, Y. Chai, L. Cui, D. Ma, Y. Fu, and A. Xiao, "A deep learning based DDoS detection framework for internet of things", IEEE International Conference On Communications, 2020.

[33] S. Potluri, S. Ahmed, and C. Diedrich, "Convolutional neural networks for multi-class intrusion detection system", 6th International Conference, MIKE 2018, Cluj-Napoca, Romania, December 20-22, 2018.

[34] Y. Ding, and Y. Zhai, "Intrusion detection system for NSL-KDD dataset using convolutional neural networks", In Proceedings of the 2018 2nd International conference on computer science and artificial intelligence, 2018, pp. 81-85.