Deep Defence

Faisal Jamil
Department of Computer Engineering
National Institute of Technology, Kurukshetra
Kurukshetra, India
faisaljamil7890@gmail.com

Under the Supervision of
Dr. Shweta Sharma
Assistant Professor
National Institute of Technology, Kurukshetra

Abstract

Nowadays, due to the extensive use and evolution of the cyberworld and the increase in the number of devices connected to the internet, the vulnerability to Distributed Denial of Service (DDoS) attacks has also grown significantly. A Distributed Denial-of-Service (DDoS) attack has become one of the fatal threats to the internet, in which attackers send a huge amount of packets to the server and restrict users from accessing the online system. DDoS attacks can overwhelm targeted systems, causing significant disruption and damage. This project aims to detect and mitigate DDoS attacks using neural networks and deep learning techniques. We utilized a dataset from Kaggle and implemented three models: Simple Recurrent Neural Network (RNN), Bidirectional Long Short-Term Memory (LSTM) , and a Hybrid model combining RNN and bi-directional LSTM. We pre-processed the dataset and created sequences for training and testing. We applied K-fold(5-fold) cross-validation to ensure robust evaluation of the models. The performance of each model was assessed based on accuracy, precision, recall, F1 score, and confusion matrix. Our results demonstrate that the deep learning models achieved high accuracy and efficiency in detecting DDoS attacks, with the Bi-directional LSTM model showing the best performance overall with an accuracy of 99.9648%. This work helps to enhance network security by providing an effective solution for DDoS attack detection.

Keywords: Deep Learning, DDoS Attack, RNN, LSTM, Cybersecurity, Network Security, Machine Learning

1. Introduction

DDoS attacks represent a significant cybersecurity concern, disrupting services by inundating targeted systems with excessive internet traffic. These attacks can cause significant financial and reputational damage to organizations. With the evolution of sophisticated intrusion techniques, attackers can quickly impair network systems. DDoS attacks are easily performed by exploiting network weaknesses and generating excessive service requests. Traditional methods for detecting DDoS attacks often struggle due to the dynamic nature of attack vectors and the escalating volume of network traffic. Consequently, there is growing interest in deep learning techniques to enhance DDoS attack detection.

Deep learning models, such as Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks, have shown promise in processing and analyzing time-series data. These models excel at learning temporal patterns and dependencies, thereby enabling the detection of anomalies in network traffic indicative of potential DDoS attacks.

Today’s society relies heavily on the internet, which is essential for economic transactions, education, and communication. However, along with its many benefits, the internet has experienced an increase in criminal activity, such as hacking, spreading false information, and denial-of-service (DoS) attacks. A DoS attack occurs when a legitimate service, system, or network is made inaccessible to its intended users. A DDoS attack, a subcategory of DoS attacks, involves an attacker breaching multiple computing systems to disrupt a specific target's regular traffic. [1]

Types of DDoS attacks include volumetric attacks, fragmentation attacks, application layer attacks, and protocol attacks. DDoS assaults are more destructive than DoS attacks because they involve several systems, making it more challenging for security teams and products to pinpoint the source of the attack. [2]

This report explores three deep learning models for DDoS attack detection: Simple RNN, Bidirectional LSTM, and a Hybrid model. We train these models using a Kaggle dataset [3], evaluate their performance through K-Fold cross-validation, and compare their effectiveness in identifying DDoS attacks.

Contributions:

- Implementation of Simple RNN, Bidirectional LSTM, and Hybrid models for DDoS attack detection.
- Utilization of K-Fold cross-validation to assess model performance.
- Comparison of models based on accuracy, precision, recall, and F1 score.
- Analysis of confusion matrices to elucidate model strengths and weaknesses.
- Proposal of a robust deep learning approach for DDoS attack detection.

2. Motivation

The increasing frequency and sophistication of DDoS attacks necessitate advanced detection mechanisms. Statistics show a year-over-year increase in DDoS attack incidents, with severe financial implications for affected organizations. According to a report, DDoS attacks larger than 250 Gbps grew by 1,300%. TCP DDoS attacks almost doubled in 2021 compared to 2020, accounting for 27% of all attacks. [4] Traditional detection methods struggle to keep up with the dynamic nature of these attacks, highlighting the need for more adaptive and intelligent solutions.

Deep learning techniques, with their ability to learn and adapt to complex patterns in data, offer a promising avenue for enhancing the effectiveness of intrusion detection systems (IDS) In 2023 alone, automated defenses mitigated over 5.2 million HTTP DDoS attacks, consisting of over 26 trillion requests. This averages to 594 HTTP DDoS attacks and 3 billion mitigated requests every hour. [5]

By developing and implementing deep learning models, we aim to significantly improve the detection and mitigation of DDoS attacks, ultimately contributing to the security and stability of network infrastructures.

3. Related Work

The increasing frequency and sophistication of Distributed Denial of Service (DDoS) attacks necessitate advanced detection mechanisms. A variety of deep learning models have been explored in the literature, leveraging diverse datasets and feature sets to enhance detection accuracy. This survey provides an overview of significant contributions in the field and identifies research gaps that warrant further investigation.

Tabl-I. Related Work on DDoS Detection

AUTHOR	Dataset	YEAR	Features	Model	Accuracy	Research Gaps

Omerah Yousuf, Roohie Naaz Mir [6]	NSL-KDD	2022	Flow statistics, Novel activation function	RNN, DALCNN	99.98%	Need for more comprehensive real-world IoT datasets, exploration of other deep learning models for better accuracy
Shurman, Yateem et al. [7]	CICDDoS2019 dataset	2020	Flow statistics	LSTM	99.919%	Limited to specific types of DDoS attacks, need for more diverse datasets to enhance model robustness
Elsaeidy, Abbas jamalipour et al. [8]	Real-life smart city dataset (Australia)	2021	Environmental, smart river, smart soil data.	Hybrid deep learning model (RBM + CNN)	99.51%	Need for more generalized models to handle a wider range of attacks and more complex real-world data scenarios
Farooq, Aqeel Baba et al. [9]	application layer DDoS dataset (Kaggle)	2021	Various network traffic features	Random Forest (RF) and Multi-Layer Perceptron (MLP)	99.5%	Challenges in real-time detection, efficiency with larger datasets, exploring ensemble

						techniques for improved accuracy
Khempetch et al. [10]	CICDDoS2019 dataset	2021	Network flow statistics, DDoS attack taxonomy	DNN, LSTM	99.90%-99.97%	Limited exploration of other neural network architecture and attack types.

RESEARCH GAPS

Despite significant progress in DDoS attack detection, existing studies often overlook the benefits of combining different deep learning architectures. Additionally, most research focuses on specific features or datasets, limiting the generalizability of the models. There is a need for hybrid models that can leverage the strengths of multiple architectures to improve detection accuracy and robustness. Insufficiently large datasets: due to the potential loss of reputation or money, the majority of victim organizations are reluctant to disclose information regarding attacks undertaken against them. Furthermore, there are no complete databases in the public domain that include all traffic kinds, including genuine, low rate, high rate, and flash traffic. [11] Insufficient effort on unknown data or zero-day attacks: when the instruction and assessment datasets contain the same traits or patterns, ML models are able to function well. However, ML-based algorithms are unable to accurately detect unknown threats in real-life situations, where attacks may be launched using novel patterns. As a result, these models must be frequently updated in order to account for novel and untested assaults [12]

4. Proposed Work

4.1 Conceptual Design Diagram

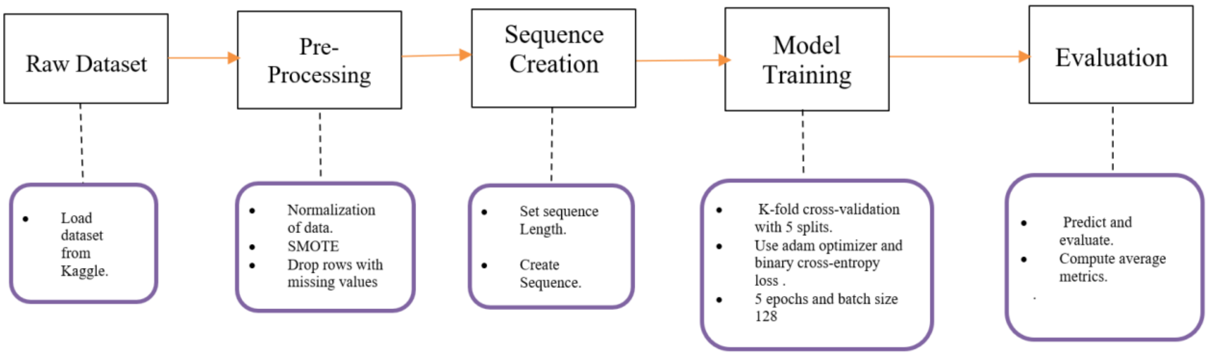

Fig-I. Conceptual design diagram

4.2 Proposed Algorithm

Algorithm 1: Hybrid Model (Simple RNN + Bidirectional LSTM)

Input: Pre-processed features and target from the dataset
Output: Average metrics across K-fold cross-validation (Accuracy, Precision, Recall, F1 Score, Confusion Matrix)
Data: Dataset split into training and testing sets

1. Σ1: 0 // Initialize lists to store metrics
2. **Data Pre-processing**:
    - Load dataset from CSV file
    - Drop any rows with missing values
    - Separate features and target label
3. **Create Sequences**:
    - for i = 0 to(len(data) - sequence_length) do
        - Extract sequences of length sequence_length end for
4. **K-fold Cross-validation**:
    - Initialize K-fold with 5 splits, shuffled and random state set
    - Initialize lists to store metrics
5. **For each fold**:
    - Split data into training and testing sets
    - **Model Definition**:
        - Create a Sequential model
        - Add Simple RNN layer with 32 units, return_sequences=True
        - Add Bidirectional LSTM layer with 64 units
        - Add Dropout layer with 0.2 dropout rate
        - Add Dense output layer with sigmoid activation
    - **Model Compilation**:
        - Compile model with Adam optimizer and binary cross-entropy loss
    - **Model Training**:
        - Fit the model on training data with validation split of 0.2 for 5 epochs
    - **Model Prediction**:
        - Predict on the test data
        - Binarize predictions at 0.5 threshold
    - **Compute Metrics**:
        - Calculate Accuracy, Precision, Recall, F1 Score, and Confusion Matrix
        - Append metrics to corresponding lists
6. **Average Metrics Calculation**:
    - Compute average of metrics across all folds
7. **Output**:
    - Print average Accuracy, Precision, Recall, F1 Score
    - Print average Confusion Matrix

Algorithm 2: Bi-directional LSTM Model

Input: Pre-processed features and target from the dataset
Output: Average metrics across K-fold cross-validation (Accuracy, Precision, Recall, F1 Score, Confusion Matrix
Data: Dataset split into training and testing sets

1. Σ1: 0 // Initialize lists to store metrics
2. **Data Pre-processing**:
    - Load dataset from CSV file
    - Drop any rows with missing values
    - Separate features and target label
3. **Create Sequences**:
    - for i = 0 to (len(data) - sequence_length) do
        - Extract sequences of length sequence_length end for
4. **K-fold Cross-validation**:
    - Initialize K-fold with 5 splits, shuffled and random state set

```
        o   Initialize lists to store metrics
5.  For each fold:
        o   Split data into training and testing sets
        o   Model Definition:
                ▪   Create a Sequential model
                ▪   Add Bidirectional LSTM layer with 50 units, return_sequences=True
                ▪   Add another Bidirectional LSTM layer with 50 units
                ▪   Add Dense output layer with sigmoid activation
        o   Model Compilation:
                ▪   Compile model with Adam optimizer and binary cross-entropy loss
        o   Model Training:
                ▪   Fit the model on training data with validation data for 5 epochs
        o   Model Prediction:
                ▪   Predict on the test data
                ▪   Binarize predictions at 0.5 threshold
        o   Compute Metrics:
                ▪   Calculate Accuracy, Precision, Recall, F1 Score, and Confusion Matrix
                ▪   Append metrics to corresponding lists
6.  Average Metrics Calculation:
        o   Compute average of metrics across all folds
7.  Output:
        o   Print average Accuracy, Precision, Recall, F1 Score
        o   Print average Confusion Matrix
```

Algorithm 3: Simple RNN Model

Input: Pre-processed features and target from the dataset
Output: Average metrics across K-fold cross-validation (Accuracy, Precision, Recall, F1 Score, Confusion Matrix)
Data: Dataset split into training and testing sets

```
1.  Σ1: 0 // Initialize lists to store metrics
2.  Data Pre-processing:
        o   Load dataset from CSV file
        o   Drop any rows with missing values
        o   Separate features and target label
3.  Create Sequences:
        o   for i = 0 to (len(data) - sequence_length) do
                ▪   Extract sequences of length sequence_length end for
4.  K-fold Cross-validation:
        o   Initialize K-fold with 5 splits, shuffled and random state set
        o   Initialize lists to store metrics
5.  For each fold:
        o   Split data into training and testing sets
        o   Model Definition:
                ▪   Create a Sequential model
                ▪   Add Simple RNN layer with 50 units, return_sequences=True
                ▪   Add another Simple RNN layer with 50 units
                ▪   Add Dense output layer with sigmoid activation
        o   Model Compilation:
                ▪   Compile model with Adam optimizer and binary cross-entropy loss
        o   Model Training:
                ▪   Fit the model on training data with validation data for 5 epochs
        o   Model Prediction:
                ▪   Predict on the test data
                ▪   Binarize predictions at 0.5 threshold
        o   Compute Metrics:
                ▪   Calculate Accuracy, Precision, Recall, F1 Score, and Confusion Matrix
```

        - Append metrics to corresponding lists

6. **Average Metrics Calculation**:
    - Compute average of metrics across all folds
7. **Output**:
    - Print average Accuracy, Precision, Recall, F1 Score
    - Print average Confusion Matrix

5. Experimental Setup

The configuration of the system to perform experiments includes a Windows 11 operating system with an Intel Core i5 64-bit processor and 8 GB of installed RAM. Google Colab was chosen for implementing and executing machine learning models due to its simplicity and ease of use. It provides a straightforward interface for writing and testing Python code and supports T4 GPU acceleration, which is essential for deep learning tasks. The main advantage of using Google Colab is that it is cloud-based, offering powerful computational resources without the need for local hardware setup. Its interactive environment is useful for iterative testing, debugging, and collaborating on code.

Training and Testing Configuration:

- **Libraries Used**: TensorFlow, Keras , Scikit-learn, Pandas, Numpy
- **Hardware**: Windows 11, Intel Core i5 64-bit processor, 8 GB RAM
- **Software**: Google Colab

Data and Data Modeling Details:

- **Dataset**: Kaggle's DDoS dataset
- **Features**: Network traffic features (e.g., packet size, duration)
- **Model Used**: Simple RNN, Bidirectional LSTM, Hybrid model

5.2 *Details of Data and Data Modeling*

- **Simple RNN:** Average Accuracy: 0.9974, Average Precision: 0.9995, Average Recall: 0.9996, Average F1 Score: 0.9974
- **Bidirectional LSTM:** Average Accuracy: 0.9996, Average Precision: 0.9996, Average Recall: 0.9996, Average F1 Score: 0.9996
- **Hybrid Model:** Average Accuracy: 0.9991, Average Precision: 0.9996, Average Recall: 0.9986, Average F1 Score: 0.9991

5.3 Experimental Results

The confusion matrix is a critical tool in evaluating the performance of machine learning models, especially in the context of detecting DDoS attacks. It provides a detailed breakdown of how well a model distinguishes between different classes (e.g., benign vs. malicious traffic). The confusion matrix consists of four main components: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN).

- **True Positive (TP)**: Instances where the model correctly identifies malicious traffic as DDoS.
- **True Negative (TN)**: Instances where the model accurately identifies benign traffic as non-malicious.
- **False Positive (FP)**: Cases where the model incorrectly labels benign traffic as malicious.
- **False Negative (FN)**: Instances where the model fails to identify malicious traffic, incorrectly classifying it as benign.

The confusion matrix provides the results as follows:

	Predicted Positive	**Predicted Negative**
Actual Positive	TP	FN
Actual Negative	FP	TN

The confusion matrix helps in understanding the performance beyond simple accuracy. By analyzing the TPs, TNs, FPs, and FNs, we can derive metrics like Precision, Recall, Accuracy, and F1-Score.

- **Precision**: The ratio of correctly predicted positive observations to the total predicted positives.

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

- **Recall**: The ratio of correctly predicted positive observations to all observations in the actual class.

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

- **Accuracy**: The ratio of correctly predicted instances to the total instances.

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

- **F1-Score**: The weighted average of Precision and Recall.

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall}$$

We use the confusion matrix for each technique used in this paper. The confusion matrices for the Simple RNN, Bidirectional LSTM, and Hybrid model are as follows:

Table-II. Simple RNN

	Predicted Positive	**Predicted Negative**
Actual Positive	25453	125
Actual Negative	8	25622

Table-III. Bidirectional LSTM

	Predicted Positive	**Predicted Negative**
Actual Positive	25568	10
Actual Negative	8	25622

Table-IV. Hybrid Model

	Predicted Positive	**Predicted Negative**
Actual Positive	25570	8
Actual Negative	34	25596

By analyzing these confusion matrices, we can better understand the strengths and weaknesses of each model in detecting DDoS attacks.

Table V below shows the accuracy, precision, recall, and F-score for the models used in Deep Defence:

Table-V. Result

Model Used	Accuracy	Recall	Precision	F-score
Hybrid	99.91%	99.86%	99.96%	99.91%
Bi-directional LSTM	99.96%	99.96%	99.96%	99.96%
Simple RNN	99.74%	99.96%	99.95%	99.74%

The above table (Table 1) shows the results after implementing our models with the dataset. The Bi-directional LSTM model provides the highest accuracy with a value of 99.96%, showcasing its superior performance in detecting DDoS attacks. The Hybrid model also performs exceptionally well with an accuracy of 99.91%, indicating its effectiveness in combining different detection techniques. The Simple RNN, while slightly lower in accuracy at 99.74%, still demonstrates strong performance.

These results highlight the effectiveness of deep learning models in detecting DDoS attacks. The lower accuracy of the Simple RNN in comparison to the other models suggests potential areas for improvement and further exploration. This discrepancy is not a drawback but rather an insightful starting point for future research and development. It highlights areas for potential growth, encourages deeper understanding, and sets the stage for future innovations in DDoS attack detection.

6. Comparison with the Existing Literature

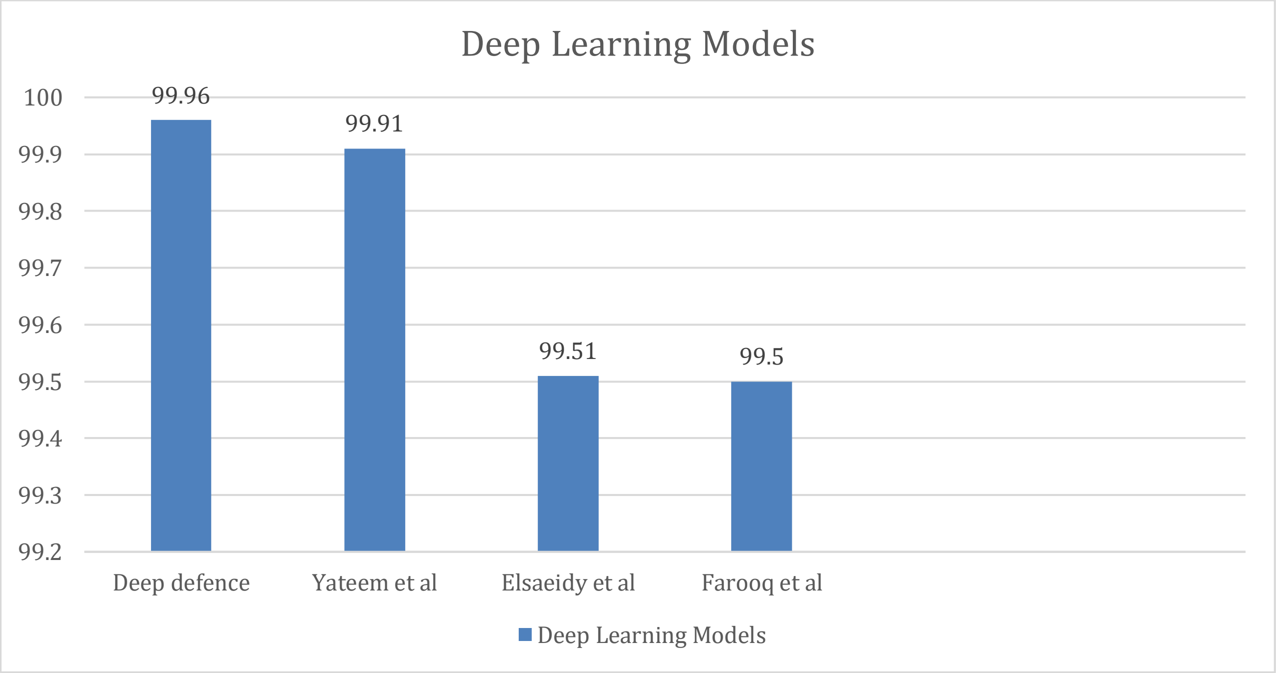

Fig-2: Comparison between different models

7. Conclusion and Future Work

This report presents an effective approach for DDoS attack detection using deep learning models, specifically Simple RNN, Bidirectional LSTM, and Hybrid models. The results demonstrate high accuracy and robustness in detecting DDoS attacks, with each model showcasing strong performance metrics:

- **Hybrid Model**: Accuracy 99.91%, Recall 99.86%, Precision 99.96%, F-score 99.91%
- **Bi-directional LSTM**: Accuracy 99.96%, Recall 99.96%, Precision 99.96%, F-score 99.96%
- **Simple RNN**: Accuracy 99.74%, Recall 99.96%, Precision 99.95%, F-score 99.74%

Compared to existing literature, our models demonstrate competitive performance, particularly in terms of precision and recall. The hybrid approach, in particular, shows potential for further improvements with optimized architectures.

Future Work

The promising results from our study indicate several avenues for future research and development:

1. **Optimized Architectures**: Further exploration of optimized hybrid models could yield even better performance metrics. This includes experimenting with different combinations of RNN, LSTM, and other advanced neural network architectures.
2. **Real-time Detection Systems**: Implementing these models in real-time detection systems will be crucial for practical applications. This involves developing efficient and scalable deployment strategies to handle live network traffic.
3. **Enhanced Feature Engineering**: Investigating additional features and more sophisticated feature engineering techniques could improve the models' detection capabilities and robustness.
4. **Adaptive Learning**: Incorporating adaptive learning mechanisms to continuously update the models based on new attack patterns and data could enhance their long-term effectiveness.
5. **Cross-dataset Validation**: Validating the models across multiple datasets and real-world scenarios will help ensure their generalizability and reliability in diverse environments.
6. By pursuing these future directions, we aim to contribute to the development of more resilient and accurate DDoS detection systems, ultimately enhancing the security of network infrastructures against evolving cyber threats.

References

[1] Cybersecurity and infrastructure Agency. Available online: **https://www.cisa.gov/uscert/ncas/tips/ST04-015**

[2] Cryptocurrency Exchange EXMO Has Been Knocked Offline by a "Massive" DDoS Attack. Available online: **https://portswigger.net/daily-swig/uk-cryptocurrency-exchange-exmo-knocked-offline-by-massive-ddos-attack**

[3] Kaggle:https://www.kaggle.com/datasets/chethuhn/network-intrusion-dataset

[4] F5: https://www.f5.com/labs/articles/threat-intelligence/2022-application-protection-report-ddos-attack-trends

[5] Cloudflare: https://blog.cloudflare.com/ddos-threat-report-2023-q4

[6] Yousuf, Omerah, and Roohie Naaz Mir. "DDoS attack detection in Internet of Things using recurrent neural network." *Computers and Electrical Engineering* 101 (2022): 108034.

[7] Shurman, Mohammad, Rami Khrais, and Abdulrahman Yateem. "DoS and DDoS attack detection using deep learning and IDS." *Int. Arab J. Inf. Technol* 17, no. 4A (2020): 655-661.

[8] Elsaeidy, Asmaa A., Abbas Jamalipour, and Kumudu S. Munasinghe. "A hybrid deep learning approach for replay and DDoS attack detection in a smart city." *IEEE Access* 9 (2021): 154864-154875.

[9] Awan, M. J., Farooq, U., Babar, H. M. A., Yasin, A., Nobanee, H., Hussain, M., ... & Zain, A. M. (2021). Real-time DDoS attack detection system using big data approach. *Sustainability*, *13*(19), 10743.

[10] Khempetch, Thapanarath, and Pongpisit Wuttidittachotti. "DDoS attack detection using deep learning." *IAES International Journal of Artificial Intelligence* 10.2 (2021): 382.

[11] Ali, Tariq Emad, Yung-Wey Chong, and Selvakumar Manickam. "Machine learning techniques to detect a DDoS attack in SDN: A systematic review." *Applied Sciences* 13, no. 5 (2023): 3183.

[12] Sabeel, U.; Heydari, S.S.; Mohanka, H.; Bendhaou, Y.; Elgazzar, K.; El-Khatib, K. Evaluation of deep learning in detecting unknown network attacks. In Proceedings of the 2019 International Conference on Smart Applications, Communications and Networking (SmartNets), Sharm El Sheikh, Egypt, 17–19 December 2019. [**Google Scholar**]