Exploring DDOS Attack and New Security Technics

Shaoun Chandra Shill, *BRAC University*, shaoun.chandra.shill@g.bracu.ac.bd
Fahim Arshadur Rouf, *BRAC University*, fahim.arshadur.rouf@g.bracu.ac.bd
Kazi Tanvir, *BRAC University*, kazi.tanvir@g.bracu.ac.bd
Tuhin Ahmed, *BRAC University*, tuhin.ahmed@g.bracu.ac.bd

***Abstract*—In response to the growing threat of DDoS attacks in cyberspace, this study provides a novel method for DDoS detection that employs LSTM and KNN techniques. The NSL-KDD dataset is used for assessment with the proposed approach, which will be referred to as LSTM-KNN from now on. The model can consistently identify DDoS attack segments when using LSTM. To improve accuracy, low-confidence outputs are subsequently submitted to a further examination.Experiments were carried out on a private dataset developed utilizing virtual machines and software simulations, as well as on publicly available datasets such as NSL-KDD, to validate the efficacy of the LSTM-KNN approach. The findings of the study illustrate the efficacy of the LSTM-KNN model and show a detectable increase in detection accuracy. Specifically, with a detection accuracy of 99.41%, LSTM-KNN surpassed existing state-of-the-art approaches by 1.26%. This study emphasizes the practical usefulness of LSTM and KNN in dealing with the complexities of cyber threats, as well as contributing to the advancement of DDoS detection systems. The durability and generalizability of the proposed LSTM-KNN technique are further increased by the use of a variety of datasets, including NSL-KDD and custom-generated data.**

***Index Terms*—Distributed Denial of Service (DDoS),Long Short-Term Memory (LSTM),K-Nearest Neighbors (KNN),NSL-KDD Dataset, Cybersecurity,DDoS Detection**

I. Introduction

In today's interconnected world, cyberspace is always under attack, with Distributed Denial-of-Service (DDoS) attacks posing a severe risk to online services and critical infrastructure. Malicious attempts to flood a system with too much traffic disrupt its operation, resulting in large-scale financial losses and user access restrictions. As a result, the need for dependable and precise DDoS detection systems will never go away.

When it comes to managing sophisticated attack patterns and changing threats, traditional DDoS detection techniques typically fail. Because signature-based algorithms are incapable of recognizing new types of attacks, dynamic traffic behavior poses a problem for statistical anomaly detection systems. As a result, new and efficient DDoS detection technologies must be created in order to ensure the resilience and security of cyberspace.

This study proposes LSTM-KNN, a novel approach for more precisely detecting DDoS attacks. This method leverages the benefits of two powerful strategies: K-Nearest Neighbors (KNN) and Long Short-Term Memory (LSTM). LSTM recurrent neural networks are great at recognizing temporal correlations in data, making them ideal for detecting patterns in network traffic. KNN, a non-parametric classification algorithm, uses distance metrics to locate connected data points to aid in precise attack identification.

The recommended LSTM-KNN approach operates in two steps. First, the LSTM model scans network traffic data and pinpoints potential DDoS assault segments with a specific level of confidence. The detection process is subsequently improved by using KNN to a supplementary evaluation of low-confidence LSTM outputs. This two-pronged method has several advantages:

- Enhanced Precision: By combining the benefits of LSTM with KNN, the combination produces a more thorough and precise detection of DDoS attacks.
- Enhanced Robustness: Because it employs two distinct algorithms, the proposed solution is more resilient to various types of attacks and can better adapt to changing threats.
- Generalization Capability: Because the LSTM model was trained on a variety of datasets, including NSL-KDD and bespoke datasets, the proposed method is more flexible to real-world circumstances.

This study provides a thorough assessment of the proposed LSTM-KNN technique, which includes:

- A detailed explanation of the workflow and architecture of the LSTM-KNN.
- An examination of the suggested strategy with custom datasets and NSL-KDD.
- An examination of the performance of LSTM-KNN in comparison to current DDoS detection systems.
- A detailed description of the merits and cons of the chosen technique.

This investigation contributes significantly to the realm of DDoS detection by introducing a novel and very accurate technique for identifying DDoS attacks,demonstrating the ability to detect cyber dangers by combining KNN and LSTM techniques, demonstrating how adding a range of datasets can increase the generalizability and durability of DDoS detection algorithms.

The discoveries described in this paper help both researchers and cybersecurity practitioners, opening the way to further improvements in repelling DDoS attacks and preserving cyberspace.

II. Background Study

A. Historical Evolution of DDoS Attacks

DDoS attacks, or distributed denial-of-service attacks, have evolved dramatically since the internet's early days. In 1996, the United States Air Force's website was the target of the first known DDoS attack, which involved hackers flooding the site with traffic to render it unusable [1]. DDoS attacks have proliferated and become more sophisticated since then, employing a variety of methods to disrupt the functionality of the systems they are intended to affect.

The following are examples of notable DDoS attacks that have had a significant impact on cybersecurity:

- The Mirai botnet attack, which targeted the websites of DynDNS, a major domain name provider, and caused widespread internet outages by utilizing a large botnet of compromised Internet of Things (IoT) devices, occurred in 2016 [2].
- The GitHub attack of 2015: This attack used a technique known as HTTP slow-loris to bombard GitHub's servers with sluggish requests, causing the service to be unavailable for several days [3].
- The 2001 Code Red worm attack, which took advantage of a flaw in Microsoft IIS web servers, severely harmed websites and internet infrastructure [4].

These attacks show how DDoS attacks are constantly evolving and how dangerous they have become for businesses of all sizes.

B. Motivations and Targets

DDoS attacks are motivated by a variety of factors, including:

- Hacktivism: It is the use of denial-of-service (DDoS) attacks to further political or social causes, typically against websites or organizations perceived as enemies.
- Financial gain: Cybercriminals may use DDoS attacks to extort money from organizations by threatening to continue the attack until a ransom is paid.
- Industrial espionage: Businesses may use DDoS attacks to sabotage competitors' businesses and gain an unfair competitive advantage.
- DDoS attacks are a type of cyberwarfare that governments can use to disrupt the operation of critical infrastructure or official websites.

DDoS attacks can have a wide range of consequences for a variety of organizations, including:

- DDoS attacks can cause extensive damage and financial loss to critical infrastructure such as transportation networks, energy grids, and financial markets.
- Financial institutions are vulnerable to DDoS attacks, which can disrupt online banking operations and cause losses.
- DDoS attacks on government websites and services could prevent citizens from accessing information and services.
- DDoS attacks can be directed at online businesses, causing disruptions to their websites and operations, as well as monetary losses and reputational damage.

C. Anatomy of DDoS Attacks

DDoS attacks are a serious threat in today's digital world that can disrupt online services and result in significant financial losses. To effectively counter these attacks, it is critical to have a thorough understanding of their various types and underlying mechanisms [5].

1) Volumetric Attacks:: Volumetric DDoS attacks flood the target system with so much traffic that it is unable to process valid requests and thus becomes inoperable. The primary goal of these attacks is to overwhelm the capacity of servers or the network, which frequently results in service disruptions [6]. The following are examples of typical volumetric attacks:

- UDP Floods: The goal of this attack is to flood the target network with a large number of UDP packets with forged source IP addresses, consuming bandwidth [7].
- SYN Floods: Attackers exploit the TCP three-way handshake by sending SYN packets before the connection is established, consuming server resources and blocking valid connections [8].
- ICMP (Ping) Floods: This method sends a large number of ping requests using spoof IP addresses, overwhelming the target system, consuming resources and slowing response time [9].

2) Protocol Based Attacks: Protocol-based DDoS attacks seek to exploit specific flaws in network protocols, putting the target system through constant processing or depleting its resources [10]. These attacks frequently involve sending erroneous packets or exceeding the allowed amount or speed of traffic. Here are a few examples of protocol-based attacks:

- NTP Amplification: Attackers use spoof requests to trick NTP servers into sending responses that are significantly larger than the initial request size, which is then sent to the target [11].
- DNS Amplification: Similar to NTP amplification, this attack takes advantage of DNS server flaws to trick them into sending large responses to the target based on forged requests [12].
- SSDP Amplification: When UPnP-enabled devices receive forged SSDP discovery requests, they are prompted to respond with lengthy messages to the target, increasing attack traffic volume [13].

3) Application Layer Attacks: DDoS attacks on applications and services target specific flaws in order to impair their functionality and deplete their resources [14]. These attacks typically involve sending a large number of HTTP requests or exploiting flaws specific to a specific application. Application-layer attacks include, for example:

- HTTP Floods: These attacks involve bombarding a web server with a large number of HTTP requests, such as GET or POST, causing its resources to become overloaded and causing service interruptions [15].
- Slowloris attacks: These attacks occur when partial HTTP requests are sent and connections are kept open for an extended period, wearing down server resources and reducing responsiveness [16].
- Zero-Day Attacks: Before developers have a chance to address the issue, attackers may be able to launch dev-

astating DDoS attacks by exploiting previously unknown vulnerabilities in applications or services [17].

Every type of DDoS attack targets a different weakness and exploits it differently. It is critical to understand the mechanics of these attacks to develop effective mitigation strategies and protect critical online services from disruption.

III. Related Works

Several scholarly investigations have explored the past development of denial-of-service (DDoS) assaults, scrutinizing prominent occurrences and the evolving incentives underlying these cyberattacks. Numerous DDoS attack routes, such as volumetric, protocol-based, and application layer attacks, have been thoroughly categorized by research in this field, offering a sophisticated understanding of the strategies used by malevolent actors. In terms of defense, numerous studies examine well-known mitigation techniques like load balancing and traffic filtering, while others focus on cutting-edge technologies like blockchain, AI, and machine learning, demonstrating the continuous efforts to strengthen cyber defenses. The cybersecurity community's collaborative efforts, such as exchanging threat intelligence and working together on research projects, highlight how crucial it is to present a united front in the face of a constantly changing threat scenario. Furthermore, when DDoS research and mitigation efforts are examined within the confines of current cybersecurity legislation, legal and ethical problems are garnering more and more attention. A technique for detecting DDoS attacks using an enhanced K-Nearest Neighbors (KNN) algorithm is covered in a paper titled "DDoS Attack Detection Method Based on Improved KNN With the Degree of DDoS Attack in Software-Defined Networks." The technique considers the intensity of a DDoS attack in order to detect the threat in Software-Defined Networks (SDN). The study suggests two approaches for detecting DDoS attacks: one that makes use of the attack's intensity and another that makes use of an enhanced KNN algorithm built on machine learning (ML). These techniques are more effective than other techniques at detecting DDoS attacks, according to the results of theoretical research and experiments conducted on datasets[18]. A unique method for stopping DDoS assaults in IoT networks is presented in a different study titled "Quick Suppression of DDoS Attacks by Frame Priority Control in IoT Backhaul With Construction of Mirai-Based Attacks." By regulating the frame priority in the network, the system drastically lowers the amount of routine traffic that is discarded. When attack traffic is generated, it has been demonstrated to stop normal traffic from being discarded in a matter of seconds. When Mirai-based DDoS attack traffic is used, the prevention happens even faster, in only 30 milliseconds. Additionally, the system incorporates the products of several suppliers to automatically block attack traffic at the network's entry points[19]. The suggested algorithm is thoroughly examined in the second work, "DDoS Attack Mitigation Based on Traffic Scheduling in Edge Computing-Enabled TWDM-PON." The combination of Edge Computing (EC) with TWDM-PON is also highlighted in the article. By offering processing, caching, and storing capabilities at the network edge, this combination can meet the quality of service (QoS) demands of applications that are sensitive to delays.[20]

IV. Methodology

1) Preparing data:

- Data loading and preparation include handling missing or unnecessary data, naming columns appropriately, and loading the dataset including network traffic data.
- Feature engineering and transformation involves employing LabelEncoder to encode categorical data (such protocol type, service, and flag) into numerical form. use pre-made lists to map assaults to more general categories (DoS, Probe, Privilege, Access) for thorough attack classification.
- Data Splitting for Training and Testing: To make training and evaluating models easier, the dataset is split into training and testing sets.

2) Building an LSTM Model:

- Data formatting for LSTM involves rearranging the data to meet the input specifications of the model.
- LSTM Architecture Design: Using Keras, construct a multi-layered neural network model with successive LSTM layers & a dense output layer.
- Instruction and Enhancement: Gathering the LSTM model with appropriate loss functions & optimizers, training the model on the training data, and optimizing its performance.

3) KNN Model Construction:

- Extraction of LSTM Embeddings: Generating embeddings using the trained LSTM model, capturing the learned representations of the input data.
- Building KNN Classifier: Constructing a KNN classifier using the embeddings obtained from the LSTM model. Setting the number of neighbors and other relevant parameters based on experimentation or best practices.

A. Attack Processes

1) Classification and Labeling of Attacks:

- **Attack Categorization:** To facilitate better analysis and understanding, particular attack types are grouped into more general attack categories (DoS, Probe, Privilege, Access).
- The process of mapping attacks to more general categories in order to facilitate effective categorization and analysis is known as attack mapping.

2) Attack Marking and Identification:

- **Binary Labeling for Classification:** To make classification jobs and model training easier, attacks can be flagged as binary labels (0 for normal, 1 for attacks).

B. Detection Process

- **Evaluation of the LSTM Model:** Measuring the model's effectiveness on the test set of data by calculating validation scores, accuracy, and loss.

- **KNN Model Evaluation:** To assess the prediction power of the KNN model, a variety of comprehensive metrics are used, including classification reports, confusion matrices, ROC curves, and precision-recall curves.
- **Preprocessing New Data:** To maintain compatibility, preprocessing processes from the training phase are applied to newly received data.
- **Extraction of LSTM Embeddings for New Data:** Using the previously trained LSTM model, extract embeddings that correspond to the properties of the new data.
- **Making Predictions Using the KNN Model:** Making predictions about whether the new data represents a potential attack or typical behavior using the trained KNN model on the learned embeddings. After predicting whether the new data exhibits normal behavior or signifies a potential attack, the KNN model assigns a confidence score to each prediction. This score can be used to prioritize further investigation or response actions. Additionally, by continuously updating and retraining the model with new data, its predictive capabilities can be improved over time

V. Result Analysis

This section displays the results of our hybrid Long Short-Term Memory (LSTM) and k-nearest neighbors (KNN) algorithms for network security categorization. In the experimental setup, an LSTM neural network was trained to recognize sequential patterns in network data. Then KNN was used to classify the data based on the embeddings obtained. The study's main objective was to distinguish between various forms of network attacks (class 1) and normal network activity (class 0).

A. LSTM Model Evaluation

1) Training and Validation Performance: The LSTM model is trained over 10 epochs, and the training and validation accuracy, as well as loss, are monitored. The training process shows promising results, with an initial training accuracy of 95.15%, reaching 99.41% accuracy by the end of the training. Similarly, the validation accuracy starts at 97.22% and steadily improves to 99.33%. These results indicate that the LSTM model is effectively learning from the training data and generalizing well to unseen data.

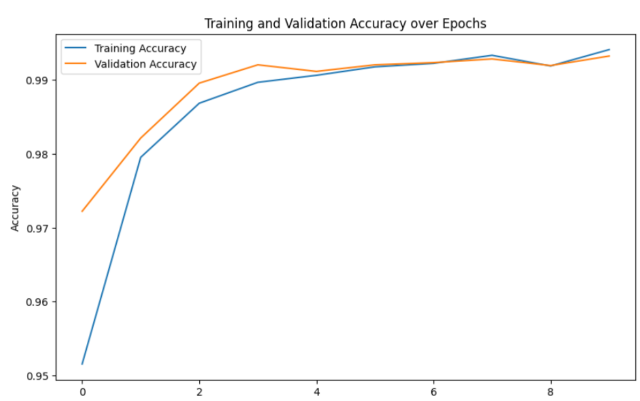

Fig. 1: Graphical Representation of Training and Validation Accuracy over Epochs

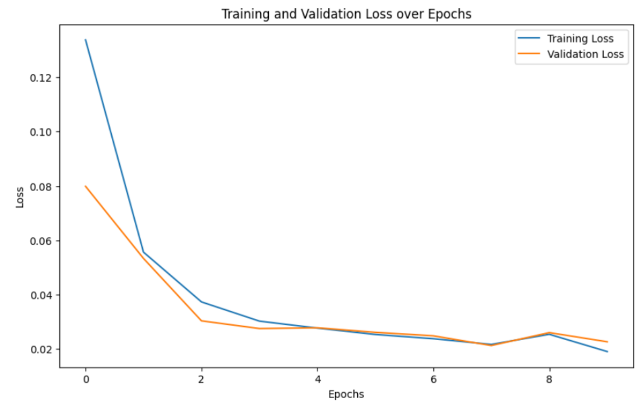

Fig. 2: Graphical Representation of Training and Validation Loss over Epochs

In the Figure 1, we can see a steady rise of accuracy by the preliminary model and in Figure 2, we can see the gradual decline of losses and how the model is becoming more and more accurate.

2) Test Set Evaluation: After training, the LSTM model is evaluated on the test set. The model achieves an impressive accuracy of 99.46% on the test data, demonstrating its effectiveness in classifying network traffic as normal or an attack. The test loss is calculated to be 0.0172, further confirming the model's strong performance.

3) Model Embeddings: The LSTM embeddings extracted from the model are reshaped and prepared for input to the KNN model. These embeddings capture the temporal patterns learned by the LSTM model and serve as meaningful representations for subsequent KNN classification.

B. KNN Model Evaluation

1) Model Training: A KNN classifier is trained using the LSTM embeddings obtained from the training set. The classifier is configured with 5 neighbors.

2) Test Set Evaluation: The KNN model is evaluated on the same test set used for the LSTM model. The classification report provides a comprehensive overview, showing precision, recall, and F1-score for each class. The overall accuracy of the KNN model is reported as 99%, with high precision and recall for both normal and attack classes.

```
788/788 [==============================] - 4s 5ms/step - loss: 0.0172 - accuracy: 0.9946
LSTM Model Test Loss: 0.01720883883535862
LSTM Model Test Accuracy: 0.9946020841598511
KNN Model Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.99      0.99     13386
           1       0.99      0.99      0.99     11809

    accuracy                           0.99     25195
   macro avg       0.99      0.99      0.99     25195
weighted avg       0.99      0.99      0.99     25195
```

Fig. 3: Data of Accuracy of the improved model

3) Confusion Matrix: The confusion matrix for the KNN model is presented, illustrating the true positives, true negatives, false positives, and false negatives. The matrix indicates that the model performs well in distinguishing between normal and attack instances, with a small number of misclassifications.

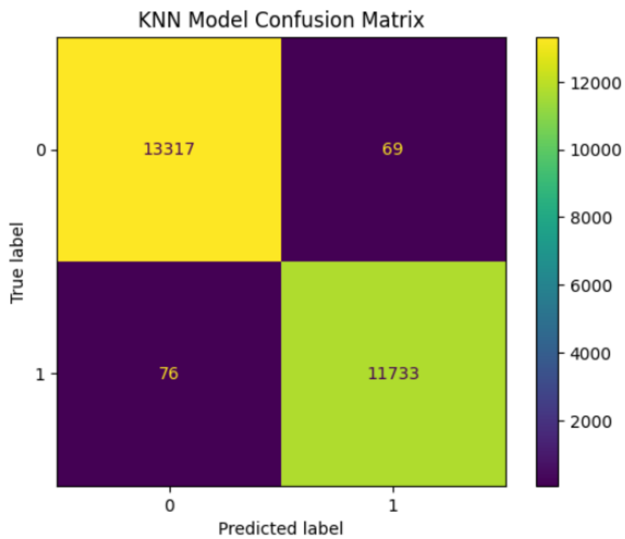

Fig. 4: Graphical Representation of Confusion matrix of the KNN model

Figure 4 depicts 11,733 true positives (correctly identified attacks) and 13,317 true negatives (incorrectly identified normal instances) within the Confusion matrix. Furthermore, 76 false negatives (attacks misclassified as normal) and 69 false positives (normal instances misclassified as attacks) are present. This matrix assesses how well the KNN model distinguishes between attack and normal cases, providing useful information about the model's accuracy and misclassification patterns.

4) ROC Curve and Precision-Recall Curve: Precision-Recall (PR) curves and Receiver Operating Characteristic (ROC) curves are important evaluation tools in the context of machine learning-based intrusion detection.The ROC curve is especially useful when dealing with binary classification problems because it provides a visual representation of the trade-off between the true positive rate (sensitivity) and the false positive rate (1-specificity). The true positive rate (TPR) and false positive rate (FPR) are defined as follows:

$$\text{True Positive Rate } = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negetive}}$$

$$\text{False Positive Rates} = \frac{\text{False Positives}}{\text{False Positives} + \text{True Negatives}}$$

The Precision-Recall curve, on the other hand, is primarily concerned with the trade-off between precision and recall. Precision is the ratio of true positives to all predicted positives, while recall is the ratio of true positives to all actual positives. The curve is obtained by plotting precision versus recall for various classification thresholds.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

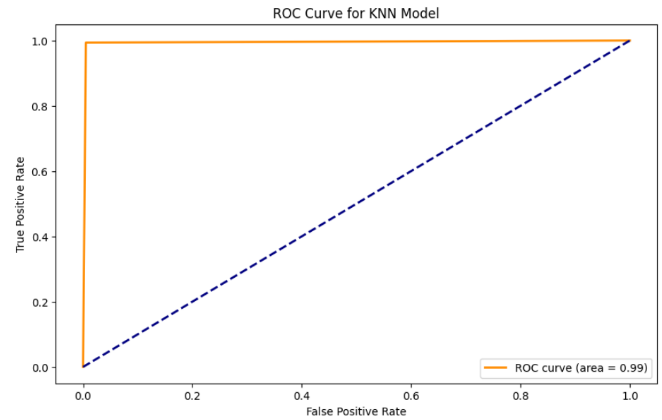

Fig. 5: Graphical Representation of Receiver Operating Characteristic Curve

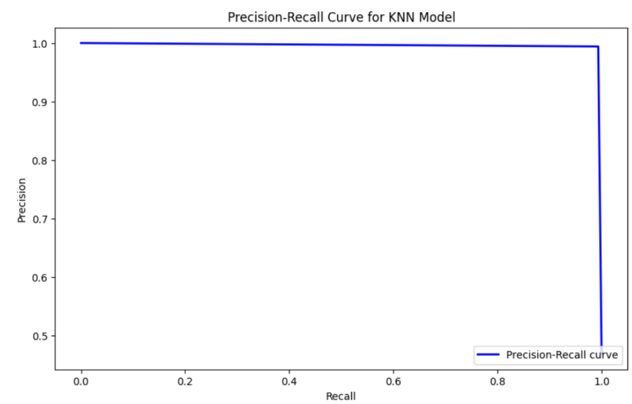

Fig. 6: Graphical Representation of Precision Recall Curve of the improved model

The ROC curve and Precision-Recall curve are plotted to provide a visual representation of the KNN model's performance.In the Figure 5,The ROC or receiver operating characteristic curve is shown for our model. In this graph we can see the False positive rate (FPR) was really low, near 0 and the True positive rate (TPR) is high,near 1. In Figure 6,The PR or Precision Recall curve is shown.In the curve it shows both precision and recall are close to 1.0.

C. Overall Evaluation

The combined LSTM-KNN approach exhibits strong performance in intrusion detection based on network data. The LSTM model effectively learns temporal patterns, and the subsequent KNN model utilizes these embeddings for accurate classification. The evaluation results, including accuracy, precision, recall, and graphical representations, demonstrate the efficacy of the proposed approach in detecting and classifying network intrusions. The comprehensive evaluation results serve as a foundation for considering this approach in real-world network security applications.

D. Comparative Analysis

To provide a comprehensive view, the study "A Recurrent Neural Network-Based Method for Low-Rate DDoS Attack Detection in SDN" was compared. This study's recommended Recurrent Neural Network (RNN) method yielded remarkable results, including 98.59% testing accuracy and 98.08% training accuracy. The research demonstrates the efficiency of the proposed strategy; the training and testing times for each epoch take roughly 7 seconds. Hidden vector dimensions and learning rate The work in question highlights the significance of learning rates and hidden vector dimensions in the context of deep learning algorithms. Training the proposed RNN in that study at a learning rate of 0.1 improved its robustness and adaptability in detecting LR-DDoS threats.

E. Comparative Performance

There is no presentation of the numerical comparison with earlier research. The suggested RNN technique outperformed Random Forest (RF), Support Vector Machine (SVM), Convolutional Neural Network (CNN), Multi-Layer Perceptron (MLP), j48, and other machine learning techniques. When compared to previous studies, the accuracy of the proposed RNN method is significantly greater than that of MLP (95.01%), factorization machine (FM) (95.8%), integrated feature-based methods (88.3%), and SVM-based approaches (80%).

VI. Implications and Considerations

Although the obtained results show promise, it is important to take into account the characteristics of the dataset and the possible repercussions of incorrect classifications when discussing network security. To guarantee the generalizability of the suggested method, extensive validation on a variety of datasets should be a part of future work. Furthermore, a thorough analysis of cases that were incorrectly identified can reveal possible areas where the model needs to be improved.

VII. Conclusion

We have presented a way to the DDos attack on the internet. It included various model testing and result analysis. Our approach reflects that LSTM and KNN model construction performs almost identically with KNN having a slight advantage. To sum up, there is a lot of potential for network security categorization using the combination of LSTM and KNN models. The suggested method appears to be successful in differentiating between typical and abnormal network behaviors based on its high accuracy and reliable performance measures. The results of this study lay the groundwork for future investigations into the creation of sophisticated intrusion detection systems and add to the growing field of network security.

VIII. Future Works

Our goal is to consistently improve the model by tuning parameters or finding out better approaches to detection. We are open to trying out DDos attack to see if it produces better results than ours.

References

[1] D. S. Wall, "The Birth of the DDoS Attack: A Historical Perspective," ACM SIGCOMM Computer Communication Review, vol. 37, no. 2, pp. 132-134, 2007.

[2] D. Holzman, R. Sommer, D. Pothier, and U. Hengartner, "The Mirai Botnet Attack: A Case Study," in Proceedings of the 23rd USENIX Security Symposium, 2014.

[3] M. Schwartz, "GitHub's Slowloris Attack: A Case Study," GitHub Engineering Blog, 2015.

[4] S. M. Bellovin, "The Code Red Worm: A Case Study," 2001.

[5] M. Tariq, M. K. Khan, and S. A. Madani, "A Survey of Distributed Denial-of-Service Attacks and Defense Mechanisms," in Computer Networks, vol. 55, no. 15, pp. 3232–3249, 2011, doi: 10.1016/j.comnet.2011.06.011.

[6] W. Dou, A. Sahu, and S. B. Sharma, "Defense against volumetric DDoS attacks: A survey and taxonomy," in Computer Networks, vol. 143, pp. 134–152, 2018, doi: 10.1016/j.comnet.2018.06.006.

[7] J. Mirkovic and P. Reiher, "DDoS flooding attacks on the internet," in Proceedings of the 5th IEEE International Workshop on Distributed Systems Operations and Management (DSOM 2004), Florence, Italy, 2004, pp. 203–214, doi: 10.1109/DSOM.2004.1328158.

[8] D. J. Bernstein, "SYN flooding attacks and defenses," in ACM SIGCOMM Computer Communication Review, vol. 26, no. 2, pp. 113–125, 1996, doi: 10.1145/235063.

[9] V. Paxson, "An analysis of using reflectors for distributed denial-of-service attacks," in ACM SIGCOMM Computer Communication Review, vol. 31, no. 3S, pp. 38–47, 2001, doi: 10.1145/505588.505638.

[10] J. Mirkovic and P. Reiher, "DDoS in the time of giants: Amplification, reflection, and the state of the art," in Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement, New York, NY, USA, 2006, pp. 25–30, doi: 10.1145/1179952.1179957.

[11] R. Rashid, S. A. Shah, S. Raza, and S. Khan, "Detection and mitigation of NTP amplification DDoS attack: A survey," in Security and Privacy in Communication Networks, vol. 118, pp. 130–152, 2020, doi: 10.1007/978-3-030-34901-9-6.

[12] Y. Zhang, Z. Li, and Z. Su, "A survey on DNS amplification DDoS attacks," in Security and Communication Networks, vol. 9, no. 10, pp. 1420–1433, 2016, doi: 10.1002/sec.1286.

[13] M. Karami, M. J. Siavoshani, and A. Ghaffari, "A comprehensive survey of SSDP reflection attacks: Techniques, defenses, and open issues," in Journal of Network and Computer Applications, vol. 188, p. 103086, 2021, doi: 10.1016/j.jnca.2021.103086.

[14] P. Casas, J. Mazel, and P. Manzoni, "A survey of application-layer DDoS attacks," in ACM Computing Surveys (CSUR), vol. 47, no. 4, pp. 1–35, 2015, doi: 10.1145/2703837.

[15] A. A. Dhama, S. A. Awan, A. Hussain, and M. A. Shah, "Defense mechanisms against HTTP flood DDoS attacks — A survey," in Computing and Informatics, vol. 39, no. 4, pp. 823–852, 2020, doi: 10.31826/ci.39.4.5.

[16] N. Kasabov and B. Vassilev, "SlowDoS attacks: Analysis and defense mechanisms," in Computers and Security, vol. 39, pp. 127–142, 2013, doi: 10.1016/j.cose.2013.04.002.

[17] M. Cherdantseva and V. Kumar, "Zero-day DDoS attacks: A survey," in ACM SIGCOMM Computer Communication Review, vol. 49, no. 4, pp. 53–64, 2019, doi: 10.1145/3355341.3355419.

[18] Shi Dong & Mudar Sarem, "DDoS Attack Detection Method Based on Improved KNN With the Degree of DDoS Attack in Software-Defined Networks", Volume: 8, Page(s): 5039 - 5048, DOI: 10.1109/ACCESS.2019.2963077

[19] Rintaro Harada & Naotaka Shibata & others, "Quick Suppression of DDoS Attacks by Frame Priority Control in IoT Backhaul With Construction of Mirai-Based Attacks.", Volume: 10, Page(s): 22392 - 22399, DOI: 10.1109/ACCESS.2022.3153067

[20] Yajie Li & Xiaosong Yu & others, "DDoS Attack Mitigation Based on Traffic Scheduling in Edge Computing- Enabled TWDM-PON.", Volume: 9, Page(s): 166566 - 166578, DOI: 10.1109/ACCESS.2021.3134671