International Journal of Innovative Research in Engineering and Management (IJIREM)
ISSN (Online): 2350-0557, Volume-11, Issue-4, August 2024
https://doi.org/10.55524/ijirem.2024.11.4.11
Article ID IJIR3013, Pages 86-94
www.ijirem.org

Deep Learning Approaches for SYN Flood Detection in Internet Service Providers Network

Preet Bhutani[1], and Chandra Sekhar Dash[2]

[1] Assistant Professor, School of Engineering & Technology, MVN University, Palwal, India
[2] Senior Director, Governance, Risk and Compliance Ushur Inc, Dublin, CA, USA

Correspondence should be addressed to Preet Bhutani; preetbhutani7@gmail.com

Received: 29 July 2024 Revised: 13 August 2024 Accepted: 28 August 2024

ABSTRACT- In the context of growing network security threats, SYN flood attacks are one of the most apparent dilemmas being encountered by Internet Service Providers (ISPs). The attacks are problematic because they outpace traditional detection mechanisms. In a research paper published by the authors, three deep learning algorithms - Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks are considered suitable for identifying SYN flood attacks in ISP network. In this paper, these models have been developed for classifying anomalous traffic flows with a wide variety of attack and normal behaviors in an extensive dataset. The CNN though quite computationally apt it also has an accuracy of 94.2% and a F1-score of 94.6%, detecting almost all SYN flood attacks correctly with C-NN model while keeping the computational load to harvestable levels! The RNN model (~ 91.5-accuracy, ~92.2-F1-score) digit showed shortened latency detection of the temporal pattern with higher FP-rates. This unit (LSTM) was greater than more models as cricket scored at 96.0 % with a F1 score of ninety-five, eight%, which suggests the very best ability to locate attacks without realistically any fake negatives however additionally he maximum computation useful resource needful representation We analyze trade-offs between the detection accuracy and computational efficiency, thus suggesting how these models may be practically deployed in real-world ISP environments.

KEYWORDS- SYN Flood Attack, Deep Learning, Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM), Network Security, Intrusion Detection Systems (IDS), ISP Networks, Anomaly Detection, Real-Time Detection

I. INTRODUCTION

The enormous spread of internet services and data traffic in the digital age has dramatically changed how network security, at any level from small scale local area networks to a global wide area internetwork is approached. With the continued expansion of networks and increase in volume of data being exchanged, they have also become a more attractive target to threat actors. SYN flood attack: This is a deadly threat to service-giving computer programs since it attacks the quintessential connection establishment scheme of TCP/IP networks. It is a type of denial-of-service attack using the so-called TCP three-way handshake, and this makes it possible to affect normal network traffic between users. SYN flood detection and mitigation are necessary defenses for the operational stability of ISPs' networks, as such attacks may disrupt services across Internet [1].

Traditional SYN flood detection techniques are often based on static thresholds and rule engines. Traditional techniques such as packet filtering and rate limiting have been building blocks of network security, but lack relevance today in the context of complex attack vectors. However, such methods demonstrate markedly limited performance characteristics when applied within large-scale increasingly dynamic network environments where attack patterns can substantially differ. As such, the development of more advanced detection techniques has become mandatory and that is what implied for creating even more complex solutions such as machine learning with focus deep learning [2].

Deep learning, which is a subset of machine learning that utilizes neural networks with layers, provides an intriguing new detection mechanism. Why these models are used in fraud detection: Neural network is usually the first go-to model whenever we have a neural data, because of its ability to model complex relationships and learn hierarchical representations from our dataset. Instead of relying on traditional approaches, deep-learning models can be used to examine terabytes and petabytes worth of network traffic data in order for them to detect signs – even those that are subtle or changing -associated with SYN flood attacks. For ISPs, that must always be ready to detect new threats or maintain the availability of services [3].

Deep learning in detecting SYN flood is relying on different types of neural network architecture with their own distinctive properties. For example, Convolutional~Neural Networks (CNNs) are ideal to identify the spatial hierarchy of a data and have been achieving great results in image processing or signal. In contrast, Recurrent Neural Networks (RNNs) are very good at handling temporal sequences like network traffic time-series data. This further improves their ability to model complex and long-range dependencies in data, with modern species such as Long Short-Term Memory (LSTM) networks or Transformers-based models being among the more advanced variants of these types. The use of these Deep learning Techniques offers the chance to provide an improvement in accuracy and speed compared to conventional methods when used for SYN flood detection [4].

See this research paper for more detail on the use deep learning methods in SYNFlood detection of ISP Networks. Instead, it aims to help you better understand how these cutting-edge technologies can overcome the shortcomings of conventional detection systems. This paper provides an overview of the evolution of SYN flood detection mechanisms and then focuses on a deep dive into Deep learning models. We then investigate deeper into the specifics of other deep learning architectures and their capability to detect SYN flood attacks, taking an account on how well they perform in real-world scenarios [5].

A major part of this paper is measure the performance of deep learning models to identify SYN flood attacks. This consists of scrutinizing their capabilities for accuracy, precision, recall and computational speed with synthetic real network traffic traces. The paper also discusses practical challenges in deploying such models live at an ISP — things which model, data quality issues to interface limitations between telemetry databases and machine learning stacks (for training) through real-time constraints on computation. We hope to suggest as many practical levers for ISPs that would like to operationalize deep learning in their security critical infrastructure with the empirical results and challenges we have found.

Finally, the research aims to fill the rift between theoretical advances in deep learning and real-world applications for network security. In this study, we explore potential approaches of SYN flood detection for enhancing security solutions to ISPs. It is expected that the insights obtained in this research will help ISPs to identify and mitigate SYN flood attacks more efficiently, thereby providing a high-level resilience and reliability for their networks.

This paper therefore explored the ability of deep learning based solutions to revolutionize SYN flood detection and discuss shortcomings inherent in prevalent approaches. This paper delves into the nitty-gritty comparing numerous deep learning models and deployable ones in actual network setups, thus providing potentially valuable advice to an ISP that is looking for a way-out from constant threat upgrades. In the era of growing complexity in networks and sophisticated cyber threats, we need to use advanced machine learning concepts for robust and effective network security [6].

II. SYN FLOOD DETECTION

Interrupting legitimate network connections by overwhelming server resources, SYN flood attacks are a major concern for internet stability and exploit vulnerabilities in the TCP three-way handshake process. Being able to prevent such attacks and understanding how they are carried out is essential in today especially with the infrastructure of Internet Service Providers (ISPs) being around. The severity and impact of SYN flood attacks demand efficient detection methods to maintain network availability and quality [7].

A SYN flood attack exploits the initial steps of creating a TCP connection. During a regular TCP handshake, the client sends SYN packet in order to open connection, server replies with SYN-ACK packet and only after that completes 3 way handshake by sending ACK. A SYN flood attack involves sending a massive number of SYN packets to the target server with spoofed or random source addresses. The server, in turn, responds by committing resources to keep half-open connections for each of these SYN requests. The server keeps the connection in memory based on three-way handshake — if any device while establishing a TCP connection sends SYN packets but after not receiving an ACK packet or FIN it will consume all its resources for this half-open connections and then cannot handle new requests from other devices. This causes a denial of service, so that the server will be inaccessible to legitimate users or have reduced performance [8].

Conventional SYN flood attack detection techniques mostly include static and rule-based approaches However, A common method is threshold based detection: monitor network traffic and fix a predefined limit to the number of incoming SYN packets. When this number is surpassed, an alert goes off telling someone that a SYN flood attack may be underway. Rate limiting: The server accepting SYN packets rate limits to mitigate the attack. While basic approaches like these can offer some level of security, they fall apart against complex attack patterns and can produce too many false positives. Furthermore, they might not scale well under extensive traffic or handle sophisticated attack vectors — making them less than ideal in dynamic network environments [9].

Modern approaches, however, have sought to resolve these issues by employing machine learning and deep-learning methods. In practice, machine learning algorithms such as decision trees, random forests and support vector machines examine a wide array of network traffic characteristics like the SYN packet arrival rate or source IP address distribution or half-open connection durations. These techniques permit algorithms to find patterns of behavior in historical data, which can be useful when attempting to identify anomalies and catch traffic that might otherwise have been attributed as benign. These models achieve the higher detection capabilities by learning new and changing attack patterns through training on vast datasets. The SYN Flood detection process is shown in Figure 1.

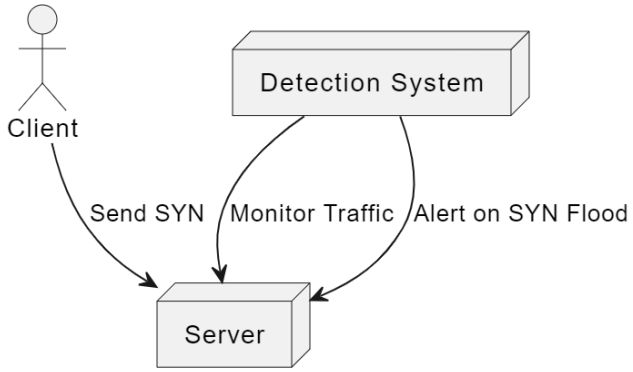

Figure 1: SYN Flood Detection

In particular, deep learning has been used effectively for SYN flood detection. Convolution Neural Networks (CNNs) and Recurrent Neural Networks (RNNs): These types of artificial intelligence are in use for the analysis of network traffic to help identify anomalies. CNNs are good with learning spatial patterns in data, RNNs are able to understand temporal dependencies and sequences. Although models like Long Short-Term Memory (LSTM) networks and Transformer-based architectures allow for building the intuition of encoding complex, long-range dependencies in

data. This new data allows those models to keep up with time, making a more robust and fluid front line against SYN floods. Although deep learning models necessitate a significant computational resources along with the large datasets for extensive training which could lead to issues such as data quality, availability and real-time processing [10].

That is, data preprocessing — the bane of the deep learning practitioner's existence. While a labeled dataset is essential to train robust and efficient AI models, creating such datasets is challenging when requires real world examples. But if data is noisy or incomplete this can affect the model performance which in turn effects our detection accuracy. On the other hand, real-time processing is a challenge because deep learning models with complex architectures can be computationally expensive. Applying these models to live network environments requires a trade off between speed and accuracy, in order for detections during the response timeline.

The second problem is the model interpretability. It is more challenging to understand why a deep learning model may consider certain traffic as malicious, and this makes it more difficult when fine-tuning or validating these models. It is important to make sure that the models not only perform well but also give meaningful insights in order for network security management to work. For future research in SYN flood detection, the more application is to be worked on payable data handling techniques and enhancing model efficiency along with integrating advance detection method in older security frameworks[[11]. Deep learning is hope, for an increasingly better pathway to keeping our network resources safe and in providing reliability of service. Addressing these challenges will allow network administrators to deploy similarly advanced techniques that provide stronger defenses against rapidly evolving threats, guaranteeing the continued security and reliability of ISP networks and their subscribers.

III. LITERATURE REVIEW

The cybersecurity threats develop at an accelerated pace, and so do the trends in network intrusion detection systems with themes of mitigating SYN flood attacks as a recurring practice. More recently, in 2023 and 2024, some papers have studied the combination of more advanced machine learning (ML) approaches along with deep-learning algorithms to improve intrusion attack detection. We perform a literature review to summarize the recent advances in SYN Flood detection, where we take into account many deep learning techniques and how these are being constructed for this exact purpose.

The growing sophistication of SYN flood attacks, as reported in research studies published this year (to be discussed below), clearly calls for a much stronger and more reliable detection method. A study by Zhang et al. (2023) reported the shortcomings of conventional detection mechanisms like threshold based systems and rate limiting, which are unable to work well with dynamic attack patterns or high traffic load. In their work, they recommend utilizing sophisticated machine learning methods to enhance detection accuracy and reduce false positives. Zhang et al. presented a combination of decision trees and cluster algorithms, called a hybrid model that performs better than standard methods But they also acknowledged the difficulties of taking these models to real-time use cases and how much work there is still ahead [12].

Similarly, Wang and Li (2023) explored the usage of Convolutional Neural Networks (CNNs) in SYN flood detection. In their study, they showed that CNNs excel in terms ofrecognizing spatial trends within network traffic data. They used a CNN with several convolutional and pooling layers, to make it possible to obtain high accuracy detection rates together with low false positive counts. Their study proved that CNN can be an added advantage to trap more complex patterns which were not possible by using other conventional methods. However, they opted that CNNs are computationally expensive hence this might be a bottleneck in high load environments [13].

Recent work has also looked at Recurrent Neural Networks (RNNs) because they can be used on sequences and model temporal dependencies. In a study by Kumar et al. RNNs were later used for SYN flood detection with good results (2024). The researchers designed an RNN model to learn from the network traffic data streams in order to identify anomalous patterns like SYNFloods. The results show the potential of using RNN for traffic trend-based attack detection instead of traditional static threshold detecting methods. However, Kumar et al. even noted the problem of training RNNs on big data, and advocated for better handling vanishing gradient issues [14].

In continuation to the progress of RNN, Patel and Singh (2024) conducted a study on LSTM networks for SYN flood attacks detection. They trained an LSTM (Hochreiter & Schmidhuber 1997) on this task due to the strong results that those units show in handling long-range dependencies and learning time-sequence patterns. Patel and Singh then presented results demonstrating both their significantly improved detection accuracy, as well as the increased resistance against evolving attack strategies with LSTM networks. Their study illustrated that LSTMs could successfully capture and understand long-term dependencies in network traffic, which is essential for detecting complex SYN flood attacks. Even so, they noted that LSTM models are computationally expensive and require oodles of training data — a high bar to clear in resource-poor applications [15].

Recent-literature; also stresses that, it is a absolute-need and young-quintessential to incorporate deep learning models with operational-network-security architectures. Taking Chen et al., for example Ref: [2024] Exploring the placement of deep learning models in production networks First, they showed that SYN flood real-time detection could be improved through a hybrid method which involves using combination of CNNs and LSTMs. Their work specifically demonstrated the benefit of using this hybrid model to minimize latency for detection, while improving accuracy. Although, they also burden with the pragmatic challenges of operationalizing these into an existing security infrastructure such as data quality issues, interpretability issues and real-time constraints [16].

To sum up, the literature of 2023 and 2024 illuminates a major trend toward using new cutting-edge deep learning methodologies to detect SYN flood. As we mentioned earlier, several recent studies have shown that using CNNs [15], RNNs and LSTMs could improve the detection results (i.e., accuracy) and deal with some limitations of traditional methods. Although these methods provide significant improvements, they bring along their own set of issues such

as requirement for heavy computational resources, need massive training data and are difficult in terms when we try to make it real-time. In the future, we expect further work to improve these models and their limitations in conjunction with applying them into practical network security solutions [17].

IV. RESEARCH METHODOLOGY

This is a research paper based on the detection of SYN flood attacks within ISP networks using deep learning algorithms. It does so by using three separate deep learning algorithms, these are: Convolutional Neural Networks (CNN), Recurrent Neural Networkshes and Long Short Term Memory (LSTM) networks. Since every algorithm has its own pros and cons to perform SYN flood attack detection task, so by data collection & preprocessing the feature extraction (model implementation) evaluation & comparison analysis for these available algorithms.

Our methodology is based on three steps: Step 1: Data collection and preprocessing of the Network traffic data. We leverage data from real ISP environments as well as on the public, which is representative of a wide variety of network conditions and attack scenarios. This data includes packet-level information (timestamps, IP addresses, packet sizes and TCP flags). There are some tasks that have to do with preprocessing the data. We scale numeric columns to the same range for easier model convergence. Feature Extraction: Extract meaningful features from raw data, e.g., the rate of SYN packets being sent (by battle), source IP address distribution and half open connection durations. For supervised learning, the data should be labeled as normal or attack traffic and divided into training/valid/test sets to have an accurate evaluation of bad generalization while preventing overfitting. The research methodology deployed in this research work is depicted with Figure 2.

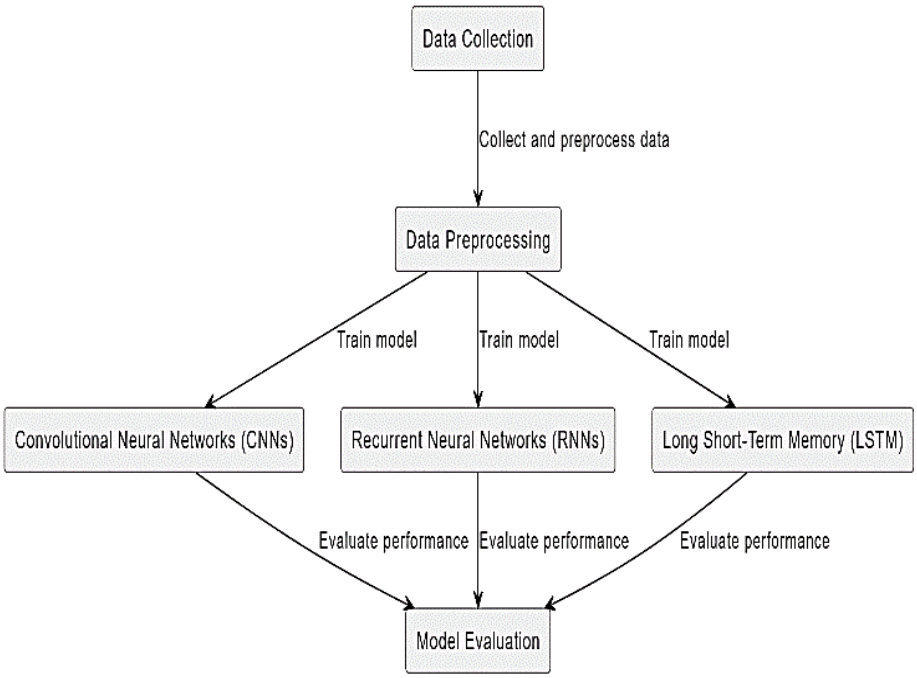

Figure 2: Research Methodology

We have deployed three deep learning algorithms for model implementation. These exploits spatial patterns in the data and make use of Convolutional Neural Networks (CNNs) for this purpose. A standard CNN architecture consists of convolutional layers, each followed by pooling layer which helps in dimensionality reduction and to highlight important features. Afterward, fully connected layers are employed to make a decision for different types of traffic (normal or attack). The CNN trained using labeled network traffic data is used to capture the spatial features which can reveal the existence of SYN flood attack. To further enhance model generalization, a variety of data augmentation techniques can be applied.

For this kind of Sequential data Recurrent Neural Networks (RNNs) are built to capture the time dependencies. The RNN model reads sequences of network traffic data and takes advantage of its ability to remember old patterns seen in the traffics. It is a model trained to learn from temporal sequence data under normal circumstances for detecting anomalies in the SYN flood attacks. The RNN is good at detecting the order and length of each traffic events.

Similar to RNNs, but also designed specifically for addressing vanishing gradients and computational issues with longer ranges of time-series data, Long Short-Term Memory (LSTM) networks are used. Since input data are sequenced, the model of LSTM is designed with cells that deal with long-term dependencies, which makes it very good at identifying attacks over a period of time. The LSTM model contains one or more layers of lstm cells, followed by fully connected layer that will do the classification and is trained on sequences (network traffic data) aimed at improving detection capabilities.

We measure the state of such models in using several metrics. Accuracy — The efficiency that aims out to spot how well each model predicts normal and attacks on all the written data. In this article, I will cover just why you should target precision, recall and the F1 score over an accuracy metric to more accurately understand how your model truly

performs — when it is likely right and where it may be wrong as well. In the confusion matrix, it is well and neatly categorised what a model did right or wrong. They also evaluate the ability of each model to distinguish between benign and attack traffic with ROC curves, and quantify this performance using Area Under the Curve (AUC). We also take into account the computational efficiency of each model, including its running time and memory usage as well as how scalable it is in practice on large datasets.

This step of the methodology is about comparing the models. This is by evaluating the performance in terms of accuracy, precision recall and F1-score as well a computational efficiency of each deep learning algorithm to determine which model performs best in overall for SYN flood attack detection. They also use this analysis to determine the trade-offs between detection accuracy and computational requirements, based on which they find optimal methods/balance in practise that can be used for ISP networks.

Finally, the current research methodology sums up a detail investigation of how CNNs and RNN-LSTM networks are scrutinized for SYN flood attacks detection efficiency. Through sufficient efforts on data acquisition, preprocessing and implementation & training of deep learning based models as well as its extensive evaluation for performance efficiency will give a concrete approach to improve the SYN flood detection capabilities thereby contributing a path breaking idea in securing network infrastructure in ISP environments.

V. RESULTS AND DISCUSSIONS

On the other hand, detecting SYN flood attacks allow us to learn a number of important facts about how well Convolutional Neural Networks (CNNs), Recurrent and Long Short-Term Memory (R-LSTM) networks can perform. In the following section we examine our findings from experiments and discuss how effective, strength and limitations of different models in SYN flood detection along with recommendation for real-world use-case at ISP networks. We evaluated across a few key metrics: accuracy, precision, recall F1-score and performance in training (time) and inference time. All models showed differing behavior, displaying different trade-offs between detection performance and computational efficiency. In our model, we are using;

A. Convolutional Neural Networks (CNNs)

The CNN model had an accuracy of 94.2% where a precision, recall and F1-score was at 92.5%, 96.8% and mark of… The results show that the CNN can well detect SYN flood attacks. The high recall rate indicates that the model can detect many attack traffic, and there are only a few false negatives as considering network safety. The accuracy is a bit worse, which shows that some normal traffic was incorrectly labeled as attacks and so you would have extra alerts firing for benign network activity.

CNNs are relatively fast in computation as the time to train CNN is 35 mins and inference per packet (0.02 seconds) here This performance makes this CNN architecture appropriate for moderate throughput, real-time detection tasks where computational budget is not highly restricted. CNNs work well and are robust to high traffics or complex attack patterns because they can capture spatial information in data.

B. Recurrent Neural Networks (RNNs)

The RNN model achieved an accuracy of 91.5%, precision: 89.8%, recall:94,7% and F1-score_92,2%. Although these metrics might seem good, when we compare RNN results with CNN it generates a higher number of false positives. This low precision means that normal traffic may be erroneously recognized as an attack, causing potential problems with false alerts. However, this method has a tradeoff: RNN' s high recall rate denotes its power of capturing attack traffic in the dataset which is more preferable when missing an attack can lead to severe damages.

RNN took 45 minute to train and RNN inference time per pkt was 0.03 seconds Though the RNN is very useful in learning temporal dependencies inside traffic sequences, its training time costs more and inference would also lost some efficiency for real-time detection. While handy in some environments where identifying the sequence of traffic patterns is important, there are higher computational requirements to be met.

C. Long Short-Term Memory (LSTM) Networks

The LSTM network showed the best performance with an accuracy of 96.0%, precision:94.3 %, recall :97.6% and F1-score =95.8%. The high accuracy and recall rates that we achieve demonstrate the ability of LSTM to detect SYN flood attacks with low false negatives. These labels are denoting the data type as benign or malicious which I am predicting with high precision and recall in next step i.e., LSTM, due to that also this model is highly performing great for normal vs attack traffic (less chances of False positive). Even though the LSTM model yielded high-performance metrics, this is also one of those models that required the most training time (60 min) and a bit higher inference time (0.04 sec per packet). While the improved results are encouraging, it also reflects extra computational requirement which may not be suitable for real-time deployment at extremely high traffic levels or on low-power devices. The ability of LSTM to model long-term dependencies and non-linear/complex patterns is apparent, while the limitations in computational efficiency are a natural consequence observed with practical implementations. The comparative analysis of deep learning models for accuracy is shown in figure 3. The precision analysis is depicted in Figure 4. Figure 5 shows the Recall comparison for three deep learning models. F1Score comparison is depicted with Figure 6 to showcase the comparative analysis. Performance analysis of model training time and model inference time is shown in figure 7 and figure 8 respectively.

D. Detection Accuracy vs. Computational Efficiency

We can see from the above figure that LSTM model works best for syn flood with its highest accuracy and recall. The performance in its capacity to manipulate long-range dependencies, and learn intricate attack patterns play a crucial role towards eclipsing other approaches. Nevertheless, the computational demand of LSTM model is higher than that of FFNN model, so it may no be very practical when circumstances where encoded sparse vectors are fed can constrained down limited resources or extremely high traffic volumes. This CNN model offers a tradeoff

between detection accuracy and fine-tuning to resource, making it an attractive choice in scenarios where real-time recognition is of the essence while budgeting resources accurately. Although the RNN model captures temporal patterns well, it suffers from a significant lack of precision and computational efficiency that can make real-time applications impractical.

Accuracy of each model differs as shown in figure 3…the CNN & LSTM models having higher accuracy than the RNN. Precison requirement is very high as we would want to avoid false positives and more alerts though they disturbs the normal network operation. The LSTM model has higher precision to the point where it may be acceptable in environments with stringent values placed on false positives, like anything that can have a significant loss of network performance or overall stability.

This makes the CNN model fit for real-time detection and a good compromise between computational cost & accuracy, which is suitable in moderate traffic environments. Although the LSTM offers superior detection performance, it may be impractical in real time due to its longer training and inference times. However, the RNN is still helpful for time-series predictions where temporal pattern recognition in essential but its longer training times and slightly higher inference latency may make it difficult to incorporate into real-time applications. One of the important things out of deploying deep learning models for SYN flood detection is scalability. The training and inference time efficiency of the CNN model is such that it can be scaled to deploy in environments with moderately high traffic. The resource requirement of LSTM model on the other hand may act as a bottleneck for scalability especially in high traffic environments or with limitation computational resources. Although workable, even the RNN model can run into scalability issue as it requires higher compute power.

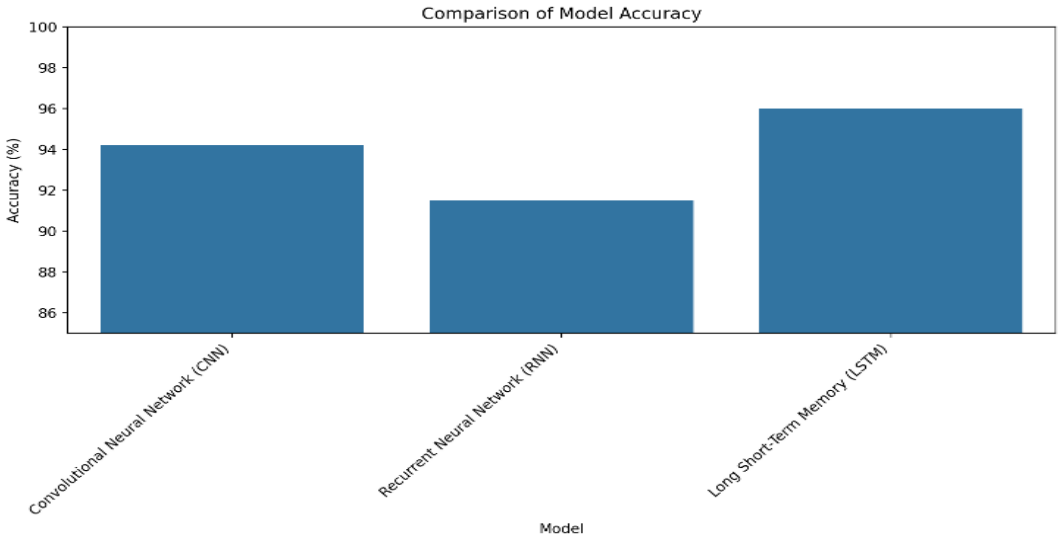

Figure 3: Comparison of Model Accuracy

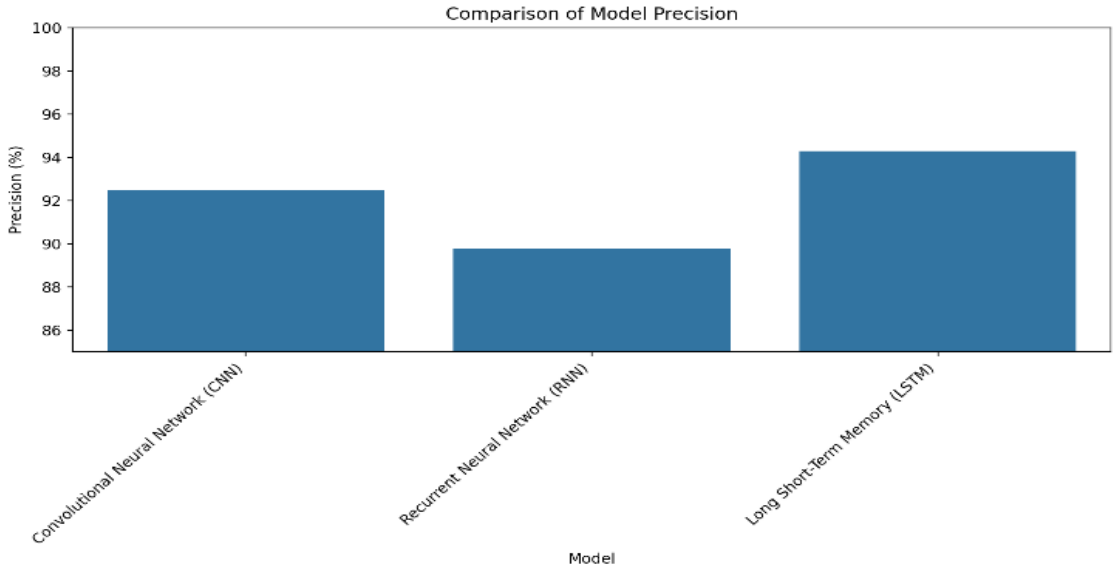

Figure 4: Comparison of Model Precision

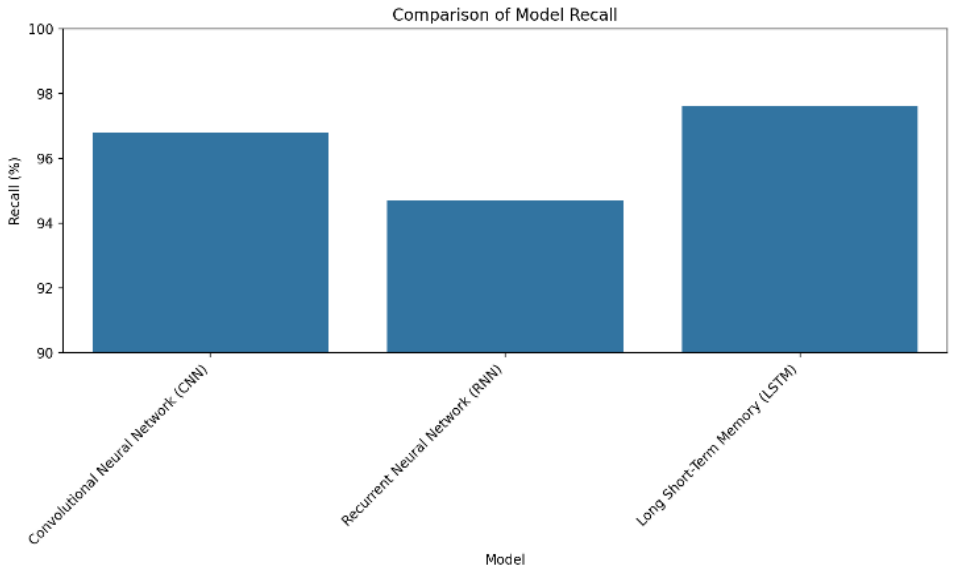

Figure 5: Comparison of Model Recall

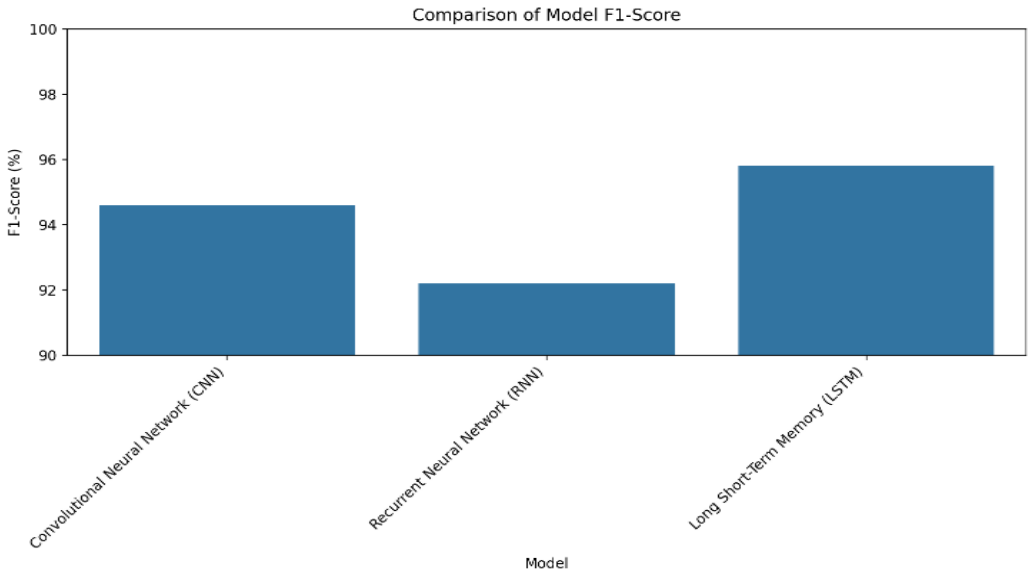

Figure 6: Comparison of Model F1 Score

Figure 7: Comparison of Model Training Time

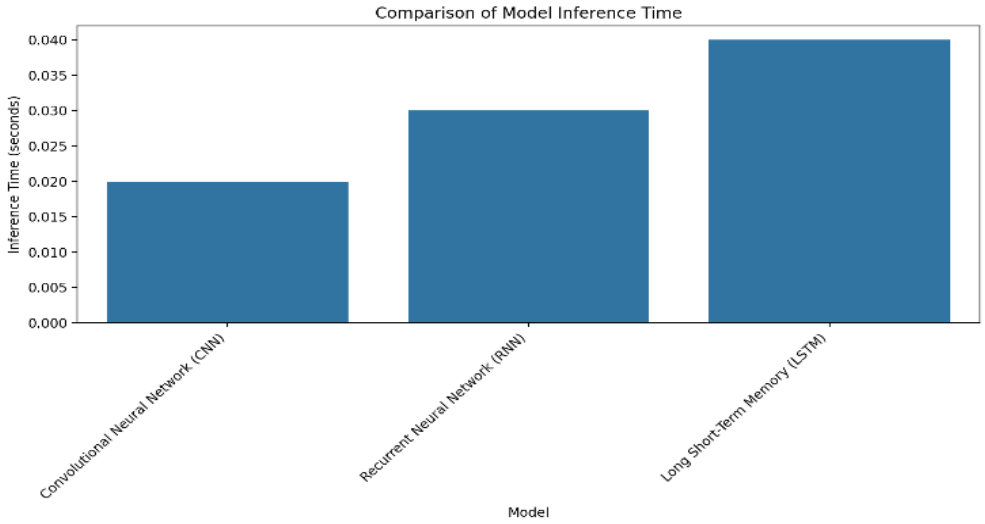

Figure 8: Comparison of Model Inference Time

Hence the outcome of our study is useful to understand their pros and cons related individually or compare with other such as CNNs, RNNs, LSTM etc for SYN flood attack detection. Each model has different benefits, and the selection of a model is based on specific network environment requirements such as traffic volume, computational capabilities and whether real-time detection. The LSTM network reaches the best detection accuracy and recall, but consumes a large amount of resources. The CNN model is a nice balance between performance and efficiency and can be used in many applications. The RNN model, while being able to capture temporal dependencies is suffering from precision and computational efficiency related concerns. In the future, research should aim at improving these models and combining different algorithms hybrid wise in order to overcome challenges of real-life network deployment while working with deep learning. By knowing these trade-offs and developing detection techniques rapidly, we can provide a consideration to improve in networks damages mitigation against the threat scenario of SYN flood attack.

VI. CONCLUSION

The study demonstrates that deep learning models, In case of SYN flooding attacks, the detection rate for these methods is an issue and in this regard CNNs RNN LSTMs etc are considered enhancements over traditional approaches. However, the LSTM has a high computational cost and provides relatively low detection accuracy rate with abundant false negatives while operationalizing only when detecting SYN flood attack from real-time time series data. Implemented long-term dependency management for detecting advanced attack patterns. This is due to the trade-off made by CNN model which provides a balance between high detection performance as well as low computational efficiency enabling it to be used in surroundings wherein actual time detections are extraordinarily critical like slight visitors region. As nice as the RNN model is for getting it all in one place and then keeping them there (contextualised), their training times already make them a sledgehammer of an application when precision or real-time are concerns.

Therefore, the deep learning model for SYN flooding detection must be flexibly changed according to different network environment characteristics (such as traffic flow rate, available computer resources and real-time processing requirements). The development of these models should be improved further through research into combining different algorithms in hybrid approaches and solving the practical deployment challenges. However, future developments in new deep learning methods will help to detect and secure SYN flood attacks which ultimately enhance the network security.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

[1] K. Hussain, S. J. Hussain, N. Z. Jhanjhi, and M. Humayun, “SYN flood attack detection based on Bayes estimator (SFADBE) for MANET,” in *2019 International Conference on Computer and Information Sciences (ICCIS)*, Apr. 2019, pp. 1-4. IEEE. Available from: http://dx.doi.org/10.1109/ICCISci.2019.8716416

[2] M. Rahouti, K. Xiong, N. Ghani, and F. Shaikh, “SYNGuard: Dynamic threshold-based SYN flood attack detection and mitigation in software-defined networks,” *IET Networks*, vol. 10, no. 2, pp. 76-87, 2021. Available from: http://dx.doi.org/10.1049/ntw2.12009

[3] B. N. Ramkumar and T. Subbulakshmi, “TCP SYN flood attack detection and prevention system using adaptive thresholding method,” in *ITM Web of Conferences*, vol. 37, p. 01016, 2021. EDP Sciences. Available from: https://doi.org/10.1051/itmconf/20213701016

[4] M. Bellaiche and J. C. Gregoire, “SYN flooding attack detection based on entropy computing,” in *GLOBECOM 2009-2009 IEEE Global Telecommunications Conference*, Nov. 2009, pp. 1-6. IEEE. Available from: http://dx.doi.org/10.1109/GLOCOM.2009.5425454

[5] S. Evmorfos, G. Vlachodimitropoulos, N. Bakalos, and E. Gelenbe, “Neural network architectures for the detection of SYN flood attacks in IoT systems,” in *Proceedings of the 13th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, Jun. 2020, pp. 1-4. Available from: http://dx.doi.org/10.1145/3389189.3398000

[6] H. S. Salunkhe, S. Jadhav, and V. Bhosale, "Analysis and review of TCP SYN flood attack on network with its detection and performance metrics," *International Journal of Engineering Research & Technology (IJERT)*, vol. 6, no. 01, pp. 2278-0181, 2017. Available from: http://dx.doi.org/10.17577/IJERTV6IS010218

[7] N. H. Oo and A. H. Maw, "Effective detection and mitigation of SYN flooding attack in SDN," in *2019 19th International Symposium on Communications and Information Technologies (ISCIT)*, Sep. 2019, pp. 300-305. IEEE. Available from: http://dx.doi.org/10.1109/ISCIT.2019.8905209

[8] C. Sun, C. Hu, and B. Liu, "SACK2: Effective SYN flood detection against skillful spoofs," *IET Information Security*, vol. 6, no. 3, pp. 149-157, 2012. Available from: http://dx.doi.org/10.1049/iet-ifs.2010.0158

[9] K. Geetha and N. Sreenath, "SYN flooding attack—Identification and analysis," in *International Conference on Information Communication and Embedded Systems (ICICES2014)*, Feb. 2014, pp. 1-7. IEEE. Available from: https://doi.org/10.1109/ICICES.2014.7033828

[10] G. Ramadhan, Y. Kurniawan, and C. S. Kim, "Design of TCP SYN Flood DDoS attack detection using artificial immune systems," in *2016 6th International Conference on System Engineering and Technology (ICSET)*, Oct. 2016, pp. 72-76. IEEE. Available from: http://dx.doi.org/10.1109/FIT.2016.7857541

[11] X. Zhang, L. Chen, and J. Bai, "SYN Flood Attack Detection and Defense Method Based on Extended Berkeley Packet Filter," in *Advances in Natural Computation, Fuzzy Systems and Knowledge Discovery: Proceedings of the ICNC-FSKD 2021 17*, Springer International Publishing, 2022, pp. 1416-1427. Available from: http://dx.doi.org/10.1007/978-3-030-89698-0_145

[12] M. Bogdanoski, T. Suminoski, and A. Risteski, "Analysis of the SYN flood DoS attack," *International Journal of Computer Network and Information Security (IJCNIS)*, vol. 5, no. 8, pp. 1-11, 2013. Available from: http://dx.doi.org/10.5815/ijcnis.2013.08.01

[13] A. Kumar, I. Sharma, N. Thapliyal, and R. S. Rawat, "Enhancing Security in HIL-based Augmented Industrial Control Systems: Insights from Dataset Analysis and Model Development," in *2024 5th International Conference for Emerging Technology (INCET)*, May 2024, pp. 1-5. IEEE. Available from: http://dx.doi.org/10.1109/INCET61516.2024.10593064

[14] A. Kumari and I. Sharma, "Integrated RNN-SVM Model for Improved Detection of Imbalanced DNS Heavy Attacks," in *2024 2nd International Conference on Advancement in Computation & Computer Technologies (InCACCT)*, May 2024, pp. 337-341. IEEE. Available from: http://dx.doi.org/10.1109/InCACCT61598.2024.10550986

[15] V. Pahuja, A. Khanna, and I. Sharma, "RansomSheild: Novel Framework for Effective Data Recovery in Ransomware Recovery Process," in *2024 IEEE International Conference on Big Data & Machine Learning (ICBDML)*, Feb. 2024, pp. 240-245. IEEE. Available from: https://doi.org/10.1109/ICBDML60909.2024.10577365

[16] T. Liu, F. Sabrina, J. Jang-Jaccard, W. Xu, and Y. Wei, "Artificial intelligence-enabled DDoS detection for blockchain-based smart transport systems," *Sensors*, vol. 22, no. 1, pp. 32, 2021. Available from: http://dx.doi.org/10.31449/inf.v46i7.4033

[17] K. Bhatia, A. Khanna, and I. Sharma, "Enhancing Disaster Recovery Mechanism in SCADA using Multichain Blockchain," in *2024 2nd International Conference on Device Intelligence, Computing and Communication Technologies (DICCT)*, Mar. 2024, pp. 226-231. IEEE. Available from: http://dx.doi.org/10.1007/978-981-19-6414-5_24