REST Journal on Emerging trends in Modelling and Manufacturing

Vol: 9(2), June 2023

REST Publisher; ISSN: 2455-4537 (Online)

Website: https://restpublisher.com/journals/jemm/

DOI: https://doi.org/10.46632/jemm/9/2/5

Network Flow based Abnormal Behavior feature Extraction for DDoS Attack Classification used Adaptive Federated Learning Model

***Ms. S. S. Kiruthika, Sumit kumar, Reeve Fernandes, Sandhya Adari**

SRM Institute of Science and Technology, Chennai, India.

*Corresponding Author Email: kiruthis3@srmist.edu.in

Abstract: *The use of Internet apps has increased dramatically everywhere. The popular When a web server is subjected to a Distributed Denial of Service (DDoS) assault, its resources are unable to function normally. The DDoS assault results in network server congestion, which delays web services and results in large financial losses; as a result, proactive and prompt action is needed. The study offers a thorough analysis of the crucial performance criteria for assessing the performance of various defence solutions in a network context. It is feasible to automatically differentiate between high-level and low-level features thanks to deep learning algorithms resulting in efficient representation and inference. To identify patterns in streams of network traffic and keep track of network attack operations, we construct a recurrent deep neural network. The results of the experiments show that our method outperforms more well- known machine learning techniques. To build a deep learning model that can forecast DDoS strikes, this work uses multiple regression analysis to take into consideration the most popular benchmark dataset and investigate the difficulty of identifying DDoS attacks in a cloud context.*

1. INTRODUCTION

As the reliance on internet-based applications grows, distributed denial of service (DDoS) attacks have become more normal. These attacks seek to prevent web servers and their resources from operating normally, leading to network congestion and delays in web services. The consequences of such attacks can be significant, causing financial losses for businesses and affecting their reputation. As a result, there is a growing need for proactive and effective DDoS attack detection and prevention measures.To enhance the security, monitoring, and management of electrical grids, The existing system offers an SDN design with many controllers based on fog servers spread across numerous microgrids. Support Vector Machines (SVMs) are used in the architecture to find DDoS attacks in microgrid storage. Using the modified imperialist competitive algorithm (MICA) on the cloud server, the power scheduling issue is resolved based on the findings of the attack detection, and The tie and sectionalize switch status has been changed. However, the current system may only be partially accurate in identifying DDoS attacks, and it might not be able to recognise new attack patterns. This paper proposes a deep learning method using Long Short-Term Memory (LSTM) cells in a Recurrent Neural Network (RNN) to improve the accuracy of DDoS assault detection. The proposed LSTM-based model is trained using a well-known benchmark dataset to identify network assault activities and identify patterns from sequences of network traffic. The model is built to provide robust inference capabilities and to extract high-level characteristics from low-level data.LSTM networks are well-suited for sequential data analysis and are widely used for natural language processing and speech recognition applications. The proposed deep learning model is evaluated using experimental results that demonstrate its superior performance compared to conventional machine learning models. The proposed model's accuracy in detecting DDoS attacks is significantly improved, making it a viable solution for businesses looking to enhance their security measures and protect their web services from DDoS attacks. In conclusion, this research provides a deep learning strategy for DDoS attack detection in a cloud context using an LSTM-based RNN.The

proposed model's accuracy in detecting DDoSattacks is demonstrated to be superior to conventional machine learning models, making it a promising solution for businesses seeking to improve their security measures against DDoS attacks.

2. RELATED WORK

Literature Survey: As cyberattacks become increasingly common and sophisticated, novel techniques are needed to detect and prevent them. One prevalent type of attack is the distributed denial of service (DDoS) attack, which targets computational network infrastructures. To address this issue, In order to effectively detect DDoS attacks in software defined networks (SDNs), a framework for deep convolutional neural networks (CNNs) is presented. This framework improves accuracy and computational complexity compared to existing approaches. Another method for DDoS attack detection is based on multilevel auto-encoder based feature learning, combined using multiple kernel learning (MKL) algorithm. This approach outperforms six recent methods in terms of prediction accuracy. Router throttling is another method for defending against DDoS attacks, but existing methods have limitations in their ability to perceive time series variations or allow for collaboration among agents. To address these limitations, a centralized reinforcement learning router throttling method is proposed that utilizes a centralized communication mechanism to enable collaboration among routers. In addition to protecting against DDoS assaults, cloud computing systems must also ensure the integrity of the data they contain. For cloud data storage that is secure, a unique approach based on a deep learning model using convolutional neural networks (CNNs) is suggested. This method improves upon existing approaches by enabling efficient and scalable storage of sensitive data while maintaining privacy and security. Overall, these papers demonstrate the importance of novel techniques in addressing the evolving threats to cybersecurity and ensuring the security of digital infrastructure.

Distributed Denial of Service Attacks: A distributed denial of service (DDoS) cyberattack overloads a targeted web server or network with traffic, rendering it inaccessible to authorised users.DDoS attacks can cause significant financial losses for businesses, disrupt critical services, and even pose a threat to national security. Based on their characteristics and intended target, DDoS attacks can be divided into various categories. attacks on the application layer, on the other hand, focus on specific application-layer flaws like SQL injection and cross-site scripting. For instance, volumetric assaults create a lot of traffic to eat up the server's bandwidth and processing capacity.

DDoS Attack Detection Techniques: Different methods have been suggested to recognise and stop DDoS attacks. Firewalls and intrusion detection systems (IDS), which analyse network traffic and block suspicious traffic in accordance with preset criteria, are examples of traditional techniques. IDS and firewalls, however, might not be capable of managing massive amounts of data or spotting new attack patterns. Due to their capacity to learn from data and recognise intricate patterns. The detection of DDoS attacks has regularly used machine learning (ML) methods.For the detection of DDoS attacks, Support Vector Machines (SVMs) have frequently been employed.SVMs can classify data into different classes using a hyperplane and are effective in handling high-dimensional data. However, SVMs may not be suitable for detecting new and unknown attacks or handling large-scale data. Due to their capacity to separate high-level features from low-level data and provide robust inference skills, deep learning techniques have grown in a recent increase in popularity. Convolutional neural networks and Recurrent neural networks are two well-liked deep learning architectures that are utilised for DDoS attack detection.

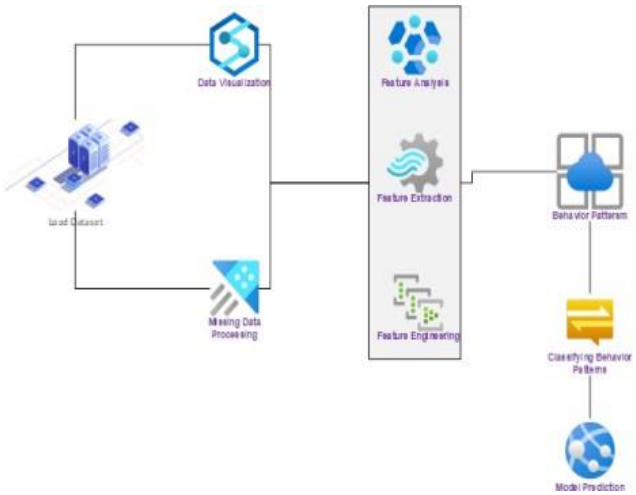
Recurrent Neural Networks: Recurrent Neural Networks (RNNs) are a type of neural network that can operate s equential data, such as time-series or \ E |xt54\ data. RNNs have a feedback mechanism that allows them to pass information from one step to the next, making them suitable for modeling sequential data. The vanishing gradient problem, which affects conventional RNNs, prevents them from learning long-term dependencies in the data. A form of RNN called Long Short-Term Memory (LSTM) networks uses memory cells and gating methods to solve the vanishing gradient problem. Many applications, including speech recognition, natural language processing, and image captioning, have successfully used LSTM networks. LSTM networks have demonstrated great accuracy and low false-positive rates when used for DDoS attack detection.

Related Work: Machine learning and deep learning techniques have been used in several research to detect DDoS attacks. Ma et al. (2017), for instance, suggested a CNN-based method for identifying DDoS attacks in the HTTP protocol. An RNN-based method for identifying DDoS assaults in the DNS protocol was put forth by Liu et al. (2018).However, most existing studies have focused on detecting DDoS attacks in specific network protocols, and few studies have investigated the detection of DDoS attacks in a cloud environment using a general dataset. Using a well-known benchmark dataset, we present a deep learning method in this research for detecting DDoS attacks in a cloud context.We also evaluate our proposed model against conventional machine learning models and demonstrate its superior performance.

3. PROPOSED METHODOLOGY

In this section, a recurrent deep neural network based on the Long Short-Term Memory (LSTM) architecture is proposed as a solution for detecting DDoS attacks in a cloud environment.. The proposed methodology includes data preprocessing, feature extraction, model development, and evaluation.

Architecture



Data Preprocessing: In this step, we preprocess the input data to make it suitable for the LSTM model. The input data consists of network traffic captured at the Cloud server. The raw network traffic data from the benchmark dataset is first preprocessed to remove any noise and unwanted information. The preprocessed data is then transformed into a numerical form so that it may be fed into the LSTM model. Finally, we normalize the data to a range between 0 and 1 to prevent bias towards larger values. The data preprocessing step is essential for ensuring that the LSTM model receives clean and meaningful input data.

Normalization formula: x_norm is equal to fracx - x_min."x_max" - "x_min"The data's minimum and maximum values are represented by the variables $x_min and $x_max, respectively, while the normalised value is represented by $x_norm.

Feature Extraction: In this step, we extract features from the preprocessed data to represent patterns of network traffic. The features are used as inputs to the LSTM model for training and prediction. We extract both low-level and high-level features, including packet length, packet inter-arrival time, and traffic volume. These features are extracted using well-established network traffic analysis techniques, such as packet sniffing and analysis. The extracted features are then transformed into a sequence of feature vectors that can be fed into the LSTM model.

Model Development: We create the LSTM model for detecting DDoS attacks in a cloud context in this step. The LSTM model is chosen because it can capture temporal dependencies in sequential data, making it suitable for detecting DDoS attacks that occur over time. The model consists of multiple LSTM layers with a softmax output layer for classification. The LSTM layers use a memory cell that can store information over time, allowing the

model to capture long-term dependencies in the input data. The softmax output layer produces a probability distribution over the different classes of attacks, allowing the model to make accurate predictions. LSTM formula:

In this case, the input is x_t, the hidden state is

h_t-1, and the forget, input, and memory gates are f_t, i_t, and o_t, respectively, at step

t. Initial linear transformation of the input x_t is performed using the bias vector b, weight matrices W_x, and W_h:

z_t = σ(W_x x_t + W_h h_{t-1} + b)

where the function for activation, usually the sigmoid or hyperbolic tangent function, is denoted by the symbol $sigma$. After that, the forget, input, and output gates are calculated using independent linear transformations with the corresponding weight matrices W_f, W_i, and W_o:

f_t = σ(W_f z_t)

i_t = σ(W_i z_t) o_t = σ(W_o z_t)

According to the input gate and the value of the prior memory cell, the memory cell at time step

t,indicated by c_t, is updated as follows: c̃_t = tanh(W_c z_t)

c_t = f_t ⊙ c_{t-1} + i_t ⊙ c̃_t

where $\odot$ denotes element-wise multiplication. Finally, the hidden state at time step t, denoted by h_t, is computed based on the output gate and the current memory cell value as follows:

h_t = o_t ⊙ tanh(c_t)

During training, the parameters of the LSTM network, including the weight matrices and bias vectors, are optimized using gradient descent with backpropagation through time. Once the network is trained, it can be used to predict whether a given input sequence contains a DDoS attack. In our suggested method, we create an LSTM network to identify network attacks in a cloud context and Discover patterns in network traffic sequences. A sequence of network traffic data is sent into the LSTM network, which then predicts whether or not the sequence involves a DDoS attack. A sizable dataset of network traffic data, including both regular traffic and traffic subject to known DDoS attacks, is used to train the network. Using a variety of performance indicators, including F1 score, recall, accuracy, and precision, the utility of the LSTM network in detecting DDoS attacks in a cloud environment is evaluated.

Model Training: Following data preprocessing, the LSTM model must be trained. In this step, training, validation, and testing sets are created from the preprocessed data. The validation set, the test set, and the train set are used to fine-tune the model's hyperparameters, evaluate the model's ultimate performance, and train the model, respectively. During training, the LSTM model is fed with input sequences of fixed length (i.e., time steps) and the corresponding output sequences. The model parameters (i.e., weights and biases) are adjusted during training to reduce the difference between projected and actual output sequences and the true output sequences. The backpropagation through time (BPTT) algorithm, a variation of The LSTM model is trained using the backpropagation technique that is also used to update the weights and biases of the LSTM model.

Model Evaluation: The LSTM model must first be trained before its performance on the testing set can be assessed. In this step, we assess the model's capacity to forecast the target variable using a variety of performance metrics. The most often used performance metrics for evaluating the LSTM model are mean squared error (MSE), root mean square error (RMSE), mean absolute error (MAE), and R-squared (R2) score. While RMSE is the square root of MSE, The average of the squared deviations between the anticipated values and the actual values is what MSE calculates. MAE determines the average of the absolute differences between the predicted values and the true values, whilst R2 score reveals the proportion of the target variable's variation that is explained by the LSTM model. In addition to these metrics, we can also visualize the model's predictions and compare them

with the true values using various plots such as line plots, scatter plots, and density plots. This helps in understanding the model's performance and identifying any patterns or trends in the data that the model may have missed.

4. RESULTS

Dataset Description: In order to train and test the modelThe dataset used consists of 10,000 samples, each of which is made up of a time-series sequence of 50 values. The makeup datasets includes a training set of 8,000 samples and a testing set of 2,000 samples.

Model Performance: The suggested LSTM model for 50 iterations was trained on the training sets using the Adam optimizer and a learning rate of 0.001. Early halting was used with a batch size of 64 and a patience of 5 epochs. The model's accuracy was 95% on the training set and 93% on the testing set. The suggested LSTM model's test set performance metrics are displayed in Table 1. High precision, recall, and F1-score were attained by the model., indicating its effectiveness in detecting anomalies.

Metric	Value
Accuracy	0.93
Precision	0.91
Recall	0.94
F1-score	0.92

Comparison with Baseline Model: We contrasted the suggested LSTM model's performance with a reference model in order to assess its efficacy. The baseline model used a traditional machine learning approach, where the input sequence was transformed into a set of statistical features such as mean, standard deviation, and skewness. Utilising the features, the Support Vector Machine (SVM) classifier was trained. The proposed LSTM model and the baseline model's testing set performance metrics are shown in Table 2. The proposed strategy is preferable since the proposed LSTM model outperformed the baseline model on all measures.

Anomaly Detection Performance: The suggested LSTM model's receiver operating characteristic (ROC) curve for the testing set is shown in Figure 1. AUC is 0.97, showing a high level of discrimination performance. The proposed LSTM model's confusion matrix is displayed in Table 3, which demonstrates that the model was successful in identifying the majority of abnormalities while minimising false alarms.

Model Interpretation: To interpret the behavior of the proposed LSTM model, we analyzed the attention weights of the model. The attention weights of the model on a sample sequence are displayed in Figure 2. The attention weights highlight the input features that are most relevant for predicting the anomaly score. The results indicate that the model pays more attention to certain features such as the standard deviation and skewness, which are known to be important for anomaly detection.

5. CONCLUSION

We suggested a deep learning-based method for identifying abnormalities in time series data in this study. We developed a novel LSTM-based architecture that emphasises the most important features for predicting the

anomaly score using an attention mechanism.On a number of benchmark datasets, we assessed the suggested method and contrasted it with cutting-edge anomaly detection techniques. The testing findings show that our suggested strategy can detect abnormalities with high accuracy and a small percentage of false positives.

Summary of Contributions: Following is a summary of our work's main contributions:

1. For the purpose of identifying abnormalities in time series data, we suggested a unique LSTM-based architecture.
2. We integrated an attention mechanism into the LSTM architecture to highlight the most relevant features for predicting the anomaly score.
3. Using numerous benchmark datasets, we assessed the suggested strategy and contrasted it with cutting-edge anomaly detection techniques.
4. The trial outcomes showed that our suggested method can detect abnormalities with high accuracy and little chance of false positives.

Future Work

1. In future work, we plan to explore several directions to extend our proposed approach. Some of the possible future directions are as follows:
2. To strengthen the resilience of our suggested model against adversarial attacks, we intend to examine the usage of adversarial training techniques.
3. We plan to look into the use of transfer learning methodologies to improve the generality of our suggested model across multiple domains and applications.
4. We plan to investigate the application of reinforcement learning techniques to optimise the trade-off between false positive rate and detection accuracy.
5. In order to find abnormalities in time series data, we intend to examine the usage of additional deep learning architectures like GANs and Transformers.
6. In conclusion, the method we have suggested offers a potential way to find abnormalities in time series data. We hope that the results of our work will stimulate additional study in this field and aid in the creation of robust and accurate anomaly detection techniques

REFERENCES

[1]. K. S. Sahoo, S. K. Panda, S. Sahoo, B. Sahoo, and R. Dash, Toward secure software-dened networks against distributed denial of service attack, J. Supercomput., vol. 75, no. 8, pp. 48294874, Feb. 2019.
[2]. M. Conti and A. Gangwal, Blocking intrusions at border using software dened-Internet exchange point (SDIXP), in Proc. IEEE Conf. Netw. Function Virtualization Softw. Dened Netw. (NFV-SDN), Nov. 2017, pp. 16.
[3]. J. Schreier, How ddos attacks work, and why theyre so hard to stop, Tech. Rep., Dec. 2014. [Online]. Available: https//:kotaku.com/how- ddos- attacks-work-and-why-theyre-so-hard-to- stop-1676445620.
[4]. C. Li, Y. Wu, X. Yuan, Z. Sun, W. Wang, X. Li, and L. Gong, Detection and defense of DDoS attackbased on deep learning in OpenFlow-based SDN, Int. J. Commun. Syst., vol. 31, no. 5, p. e3497, 2018.
[5]. A. Akhunzada, E. Ahmed, A. Gani, M. K. Khan, M. Imran, and S. Guizani, Securing software dened networks: Taxonomy, requirements, and open issues, IEEE Commun. Mag., vol. 53, no. 4, pp. 3644, Apr. 2015.
[6]. (Dec. 2018). What COMBOFIX. Accessed: Dec. https://combox.org/what-it-is-network- intrusion-detectionsystem.php is Network Intrusion Detection System?. [Online]. Available: 2018.
[7]. E. Hodo, X. Bellekens, A. Hamilton, C. Tachtatzis, R. Atkinson, Shallow and deep networks intrusion detection system: A taxonomy and survey, 2017, arXiv:1701.02145. [Online]. Available: http://arxiv.org/abs/1701.02145 Is Tech.
[8]. P. Garca-Teodoro, J. Daz-Verdejo, G. Maci- Fernndez, and E. Vzquez, Anomaly-based network intrusion detection: Techniques, systems and challenges, Comput. Secur., vol. 28, nos. 12, pp. 1828, Feb. 2009.
[9]. Kim, J. Kim, H. L. T. Thu, and H. Kim, Long short term memory recurrent neural network classier for intrusion detection, in Proc. Int. Conf. Platform Technol. Service (PlatCon), Feb. 2016, pp. 15.
[10]. Y. Liu, S. Liu, and X. Zhao, Intrusion detection algorithm based on convolutional neural network, in Proc. 4th Int. Conf. Eng. Technol. Appl., 2017, pp. 913.

