CLSTMNet: A Deep Learning Model for Intrusion Detection

Ahmed Sardar Ahmed ISSA¹, Emrullah Sonuç²

Computer engineering
Karabuk University
Karabuk, Turkey
Computer engineering
Karabuk University
Karabuk, Turkey

zalbayrak@karabuk.edu.tr ahmedserdar88@gmail.com

Abstract. Intrusion detection as well distributed denial of service (DDoS) are vital in ensuring computer network security. Some researchers claim that current approaches cannot meet the requirements of today's networks are either not workable or sustainable. In a more specific sense, these concerns are related to an increasing number of human interactions, along with reducing levels of detection ability. With our study, a novel deep learning model for intrusion detection is developed for addressing these issues. We proposed a novel deep learning classification algorithm constructed using Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) named CLSTMNet. Our proposed model has been implemented and evaluated using the benchmark NSL-KDD datasets. Compared with many conventional machine learning algorithms, the satisfied outcomes have been obtained from our model.

Keywords Deep Learning, CNN, LSTM, Intrusion Detection, DDoS.

1 Introduction
The Internet is getting increasingly exposed to different types of cyber-attacks today. Security hackers are inventing new ways to evade protective measures on a daily basis to avoid detection [1]. A Denial of Service (DoS) attacks deny and reject access to shared resources for users that are legitimate. A DDoS (distributed denial of service) attack regularly involves numerous computing resources and targets to conduct a coordinated denial-of-service attack against one or several targets [2]. It is generally directed toward system resources and network bandwidth, which range or assortment from the Network layer to the way up to the Application layer. In the meantime, 1999 was the year of the first DDoS attack [3]. DDoS has reached the point of criticality, has gained widespread prominence, and is continuing to evolve at a rapid speed.

DDoS detection is an important step in any DDoS protection mechanisms. Even so, it's indeed nearly impossible to identify DDoS attacks because they almost always resemble authorized traffic. An attack activity with insufficient traffic look like a nonthreat, but can soon become an overwhelming demand [4]. The techniques of statistical machine learning are widely employed by researchers to identify DDoS attacks. Machine learning methods seem to be doing a better job than statistical methods at detecting DDoS attacks. Even so, they are subject to various problems, which include: 1) necessitating comprehensive network experience and DDoS experiments to determine the appropriate statistical features; 2) restricted to a single or a few DDoS attack vectors; 3) To keep up with improvements in
systems and attack vectors, the model and threshold value must be updated; 4) vulnerable to slow attack rate [5].

Our lives are already feeling the effects of machine learning techniques that are changing thanks to rapid advances in this area. We see it as a fundamental component of information security as well as an approach to changing the terrain. There have been many advancements in the use of machine learning intrusion detection techniques over the past decade with the goal of increasing adaptability and accuracy. However, such methods are challenged by an inherent dependency on prior knowledge, inefficiency and inability to learn with large amounts of data, and they also do not typically exhibit strong learning potential. The recent deployment of deep neural networks for the recognition of problems with detection has greatly enhanced their odds of success. [6].

It doesn't require much in the way of machine learning or human knowledge to apply deep learning algorithms accelerated through the availability of hardware processing and big data, the intricacy of the data can be parsed from raw inputs [7]. Several recent researches have shown that deep learning-based IDS can perform better conventional ML in terms of recognition capability. To generalize performance in an adversarial environment, we have to also be concerned that an adversary will adopt an abundance of deep learning strategies. A study may have any number of valid and invalid conclusions, but it is preferable to have a small number of valid conclusions or assumption and a larger number of incorrect ones [8]. For instance, adversaries can input data with insignificant differences with the purpose of disrupt or misrepresent the learning process to defeat detection of attacks, or induce and bring about a harmless or innocent input to be interpreted as an attack.

Using deep learning, we have designed and implemented a novel model which can effectively identify DDoS attacks. The model we propose is a combination of Convolutional neural network (CNN) along with Long Short-Term Memory neural network (LSTM). This novel hybrid model achieves outstanding results when applying on NSL-KDD dataset.

Other sections of the paper are arranged out as follows. Section II deliberate about and conclude the related work. Section III discuss the methodology of our proposed model. Section II is concluded by discussing the NSL-KDD dataset and our results in comparison and evaluation with conventional machine learning techniques. Finally, Section V the conclusion of our work.

2 Related work

Results indicate that the approaches of the deep learning are advantageous in dealing with issues like feature representation and selection, as well as data label problems to construct an effective Intrusion Detection System (IDS). As instance, Shone et al. [9] proposed a new deep learning classification model Non-Symmetric Deep Auto-Encoder (NDAE) method to learn feature for unsupervised problem. Yin et al. [10] for intrusion detection they propose a deep learning approach using Recurrent Neural Networks (RNN-IDS). Furthermore, they use multi-class and binary classification scenarios for testing the performance of the model's accuracy, checking the performance of each group of neurons as well as varying the learning rate. Javaid et al [11] suggested that we could rely on a more complicated deep learning-based approach for implementing a scalable and flexible Network Intrusion Detection System (NIDS). They apply and implemented a sparse autoencoder and soft-max regression-based. Tang et al. [12] flow-based anomaly detection has been used to incorporate advanced deep learning capabilities. Nevertheless, little effort has previously been made to examine the potential dangers of adversarial deep learning in the IDS system. The recent vulnerability discovered limits the use of deep learning models in safety- and security-critical control systems like voice-controlled controllers and intrusion detection systems, as well as overall automated information security assessments.

Niyaz et al. [13] They are suggested a system which utilizes deep learning for DDoS detection. For the assessments, the experiments use customized traffic traces. You et al. [14] design and implement an automatic security auditing and vulnerability scanning solution for short messages (SMS), their strategies are constructed on an RNN (Recurrent Neural Network). Wang et al. [15] advocate for some kind of approach to discovering malicious JavaScript. The proposed approach uses a 3-layer Stacked de-noising Auto-encoder (SdA) with linear regression. The system outperformed other classifiers, resulting in a true positive prediction but also to the second highest number of false positives. The work by Hou et al. [16] designated the creative approach to detecting adware in their commercial app/services detection system
called Deep4MalDroid. Their process utilizes stacked auto-encoders, with the most accurate results found from having three layers. When they used the 10-fold cross validation, their approach had a greater percentage of correct detection rate. Lee et al. [17] introduce and implement a robust deep learning to monitoring in semiconductor manufacturing. The SdA approach was implemented for providing an unsupervised learning solution. Comparing with conventional methods demonstrated an accuracy increase of 14%.

Even with the excellent detection rates found, there is still room for improvement. Such vulnerabilities and liabilities embrace the requirement for operators, long and complex training and levels of imprecision in training datasets. As of yet, most researchers are still experimenting with different algorithm and techniques to develop an efficient and accurate solution for a specific dataset, the problem, the area is in an infantile stage. Since the results presented in this paper will be applicable to this particular pool of knowledge, we believe it will be of value to the current research.

3 METHODOLOGY

The major differences between Deep learning and classical machine learning are that the relevant features of the input are manually created, and that the output of the machine learning model is automatically determined via mapping relevant features to the values. The more layers in deep learning, the more complicated the system becomes. The basic features are discovered by themselves, which produce different levels of output, which result in many different qualities of various combinations. Each level shows abstract characteristics most of which are found from the characteristics of the previous level. In our study, we first apply a preprocessing technique on the entire suggested dataset, the we constructed a deep learning structure that consist of CNN and LSMT.

A. Preprocessing

It is commonly believed that data preprocessing is an important and significant part of machine learning and data mining. Preprocessing may be used to manage numerous challenges in a large dataset in order to produce finer and better results. To ensure that all desired data fits and is properly and is expanded, specific data preprocessing techniques should be employed [18]. One of the preprocessing techniques is StandardScaler which Standardize features by removing the mean and scaling to unit variance. The standard score of a sample x is calculated as:

\[ z = \frac{x - \mu}{\sigma} \]

where \( \mu \) is the mean of the training samples or zero if with_mean=False, and \( \sigma \) is the standard deviation of the training samples or one if with_std=False. By evaluating the samples in the training set, centering and scaling take place independently. \( \mu \) and \( \sigma \). Also, mean and standard deviation are recovered so that they can be applied to future data.

B. CLSTMNet Architecture

The overall CLSTMNet structure is shown in “Fig. 1”. Which is the combination of CNN and LSTM.

![Figure 1 CLSTMNet Structure](image)

We can use CNNs to represent complex patterns above them in a given dataset by utilizing simple patterns. CNNs are unusual multilayer neural networks, perhaps the most elaborate neural networks there has ever been. It uses a similar backpropagation algorithm for most neural networks. The difference between CNNs and other algorithms is in the architectural design. There are a few components of a CNN architecture, one of which is an input layer, multiple hidden layers, and another layer of which is the output. Convolution layers, pooling, and full-connected layers are the three main parts of hidden layers in
CNNs. The input data is generated by the kernel and the resulting data is applied to the filter; this process is called convolution. The convolution layer processes subsample the output of the Pooling layer. The main objective is to reduce dimensionality.

LSTM is a special type of RNN which are designed specifically on the way to gather up long-term dependencies. A LSTM network has a similar structure to an RNN. The primary difference between LSTM and basic unit of RNN is that the latter includes a memory block. The three gate functions that represent information processing are used to regulate the way information channels that are found in the memory block. Recurrent memory can be represented as a weight-based associative array. After some initial time of training, general knowledge about the data is gained. Similarly, it relies on short-term memory and links from every single node to the next. At this point, the middle type of memory cell is denoted by LSTM technique. It is called a compositional memory cell when it is constructed from ordinary nodes in a specific procedure that include the multiplex nodes.

The general structure of our proposed CLSMTNet is like that:

- First layer is input layer which is the preprocessed data from the used dataset.
- Second layer is one-dimensional convolutional layer with ReLU activation function, 3 kernel size, 10 filter, stride by 1.
- Third layer is one-dimensional max polling layer with 2 pool size.
- Fourth layer is one-dimensional convolutional layer with ReLU activation function, 3 kernel size, 10 filter, stride by 1.
- Fifth layer is one-dimensional max polling layer with 2 pool size.
- Sixth layer is LSTM layer with ReLU activation function.
- Seventh layer is output layer, which is fully connected layer with Softmax activation function.

C. Learning

For initializing all weights the glorot_uniform initializer [19] are used. This glorot_uniform is a great way to get samples from a uniform distribution within [-limit,limit]. The limit is sqrt(6/(fan_in+fan_out)) where fan_in is the number of input data or nodes in the weight tensor while fan_out is the number of next or output nodes in the weight tensor. The training phase is to update weights. Backpropagation is the approach used in this training phase. As a loss function the Sparse Categorical Cross-entropy is used to calculate and generate the error. ADAM: a stochastic gradient descent optimizer [20] is used for updating the weights with the learning rate (0.0001) and with a batch size of 32. The learning rate is the most important hyper-parameter to change. Convergence of the training can result in an excessively high learning rate. By keeping the learning rate small, algorithms get caught in a local minimum. The networks were trained for 500 epochs. The word epoch refers to a complete period of training overall samples.

4 Results

Our proposed model has been evaluated by using the NSL-KDD dataset. The KDD’99 test dataset from the ARPA’98 IDS Evaluation is frequently employed for this type of testing. Some problems were found with the KDD’99 dataset [21] like, Due to dataset imbalance, some classes of attacks are identified too easily in the training and test data. The NSL-KDD dataset was enhanced and improved over the KDD’99, in that all the unnecessary and redundant records were removed and the records of each class were reassigned, making it more relevant to algorithm performance benchmarks. Each dataset record in the NSL-KDD contains 41 attributes.

For evaluating the performance of our proposed models, the accuracy metrics have been applied. The accuracy of our proposed model was very satisfying. We compared our results with the result in [21] from different machine learning algorithms like Multi-layer Perceptron, Random Tree, Random Forest, Naive Bayes (NB), NB Tree and J48 as illustrated in Table 1 and “Fig. 2”. In [21], the researchers uses different algorithms with a full training and testing the whole features.
Table 1 Accuracy Comparison Of Different Algorithms

| Algorithm                  | Accuracy (%) |
|----------------------------|--------------|
| Our proposed CLSTMNet      | 99.28        |
| Support Vector Machine (SVM)| 69.52        |
| Multi-layer Perceptron     | 77.41        |
| Random Tree                | 81.59        |
| Random Forest              | 80.67        |
| NB Tree                    | 82.02        |
| Naive Bayes (NB)           | 76.56        |
| J48                        | 81.05        |

Figure 2 Accuracy Comparison Of Different Algorithms

Conclusion
In this paper, an efficient deep learning architecture have been designed and proposed called Convolutional Long-Short Term Memory (CLSTMNet) for intrusion detection and DDoS classification. Our proposed model is a combination from CNN and LSTM, also a preprocessing technique have been used for our data which is considered a very important phase for any data. We showed a state-of-the-art performance on the most challenging dataset NSL-KDD when comparing to other Classifiers with accuracy 99.28%.

References
[1] O. M. Ahmed and A. B. Sallow, “Android Security: A Review,” Acad. J. Nawroz Univ. Vol 6 No 3, 2017, doi: 10.25007/ajnu.v6n3a99.
[2] X. Yuan, C. Li, and X. Li, “DeepDefense: identifying DDoS attack via deep learning,” in 2017 IEEE International Conference on Smart Computing (SMARTCOMP), 2017, pp. 1–8.
[3] P. J. Criscuolo, “Distributed denial of service: Trin00, tribe flood network, tribe flood network 2000, and stacheldraht ciac-2319,” California Univ Livermore Radiation Lab, 2000.
[4] R. R. Zebari, S. R. M. Zeebareae, and K. Jacksi, “Impact Analysis of HTTP and SYN Flood DDoS Attacks on Apache 2 and IIS 10.0 Web Servers,” in 2018 International Conference on Advanced Science and Engineering (ICOASE), 2018, pp. 156–161.
[5] J. Mirkovic and P. Reiher, “A taxonomy of DDoS attack and DDoS defense mechanisms,” ACM SIGCOMM Comput. Commun. Rev., vol. 34, no. 2, pp. 39–53, 2004.

[6] K. B. Obaid, S. R. Zeebaree, and O. M. Ahmed, “Deep Learning Models Based on Image Classification: A Review,” Int. J. Sci. Bus., vol. 4, no. 11, pp. 75–81, 2020.

[7] O. Ahmed and A. Brifcani, “Gene Expression Classification Based on Deep Learning,” in 2019 4th Scientific International Conference Najaf (SICN), 2019, pp. 145–149, doi: 10.1109/SICN47020.2019.9019357.

[8] S. R. Zeebaree, O. Ahmed, and K. Obid, “CSAERNet: An Efficient Deep Learning Architecture for Image Classification,” in 2020 3rd International Conference on Engineering Technology and its Applications (ICETE), 2020, pp. 122–127.

[9] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, “A deep learning approach to network intrusion detection,” IEEE Trans. Emerg. Top. Comput. Intell., vol. 2, no. 1, pp. 41–50, 2018.

[10] C. Yin, Y. Zhu, J. Fei, and X. He, “A deep learning approach for intrusion detection using recurrent neural networks,” IEEE Access, vol. 5, pp. 21954–21961, 2017.

[11] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, “A deep learning approach for network intrusion detection system,” in Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS), 2016, pp. 21–26.

[12] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, “Deep learning approach for network intrusion detection in software defined networking,” in 2016 international conference on wireless networks and mobile communications (WINCOM), 2016, pp. 258–263.

[13] Q. Niyaz, W. Sun, and A. Y. Javaid, “A deep learning based DDoS detection system in software-defined networking (SDN),” arXiv Prepr. arXiv1611.07400, 2016.

[14] L. You, Y. Li, Y. Wang, J. Zhang, and Y. Yang, “A deep learning-based RNNS model for automatic security audit of short messages,” in 2016 16th International Symposium on Communications and Information Technologies (ISCIT), 2016, pp. 225–229.

[15] Y. Wang, W. Cai, and P. Wei, “A deep learning approach for detecting malicious JavaScript code,” Secur. Commun. Networks, vol. 9, no. 11, pp. 1520–1534, 2016.

[16] S. Hou, A. Saas, L. Chen, and Y. Ye, “Deep4maldroid: A deep learning framework for android malware detection based on linux kernel system call graphs,” in 2016 IEEE/WIC/ACM International Conference on Web Intelligence Workshops (WIW), 2016, pp. 104–111.

[17] H.-W. Lee, N. Kim, and J.-H. Lee, “Deep neural network self-training based on unsupervised learning and dropout,” Int. J. Fuzzy Log. Intell. Syst., vol. 17, no. 1, pp. 1–9, 2017.

[18] F. S. Zulkepli, R. Ibrahim, and F. Saeed, “Data Preprocessing Techniques for Research Performance Analysis,” in BT - Recent Developments in Intelligent Computing, Communication and Devices, 2017, pp. 157–162.

[19] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in Proceedings of the thirteenth international conference on artificial intelligence and statistics, 2010, pp. 249–256.

[20] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014.

[21] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, “A detailed analysis of the KDD CUP 99 data set,” in 2009 IEEE symposium on computational intelligence for security and defense applications, 2009, pp. 1–6.