International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 10 Issue: 10 | Oct 2023 www.irjet.net p-ISSN: 2395-0072

A Comparative Study of Deep Learning Approaches for Network Intrusion Detection in Cybersecurity

Rishabh Bhandari[1], Udit Rungta[2]

[1]*Student, School of Information Technology and Engineering (SITE), Vellore Institute of Technology, Vellore, India.*
[2]*Student, Department of Computer Science and Engineering, Amity University, Noida, Uttar Pradesh, India.*

Abstract - *Due to the increasing emphasis on cyber safety in the contemporary world and the uncertainty and sophistication of various cyber attacks, an Intrusion Detection System (IDS) has become crucial in all the newest ICT frameworks. This importance arises from the necessity to discern the nature of these attacks, prompting the integration of Deep Neural Networks (DNNs) into IDS for enhanced security measures. This paper employs DNNs to forecast the attacks on Network Intrusion Detection Systems (N-IDS), utilizing a learning rate of 0.1 and is executed over 1000 epochs on the 'KDD Cup-99' dataset for both training and benchmarking purposes. To ascertain the efficacy, the dataset was also trained using several traditional machine learning algorithms and DNNs with layers varying from 1 to 5 for comparative analysis. The outcome of this research indicates that a DNN with 3 layers demonstrates superior efficacy in comparison to the other classical machine learning algorithms and varying layers of deep learning models.*

Key Words: **Intrusion detection, deep neural networks, machine learning, deep learning, DARPA dataset**

1. INTRODUCTION

In the modern world, fast-paced technological advancements have encouraged every organisation to adopt the integration of information and communication technology(ICT). Hence creating an environment where every action is routed through that system that leaves the company open to attack should the ICT system's security be breached. Therefore, this calls for multilayered detection and protection schemes that can handle truly novel attacks on the system as well as able to autonomously adapt to the new data.

There are multiple systems that can be used for shielding such ICT systems from vulnerabilities, namely anomaly detection and IDSs. The difficulty of creating rules for anomaly-detection systems is one of their flaws. Each protocol being analyzed must be defined, implemented and tested for accuracy. Another pitfall relating to anomaly detection is that harmful activity that falls within the usual usage pattern is not recognized. Therefore the need for an ID that can adapt itself to the recent novel attacks and can be trained as well as deployed by using datasets of irregular distribution becomes indispensable.

Intrusion Detect Systems (IDSs) are a range of cyber-security-based technology initially developed to detect vulnerabilities and exploits against a target host. The sole use of the IDS is to detect threats. Therefore it is located out-of-band on the infrastructure of the network and is not in the actual real-time communication passage between the sender and receiver data. Instead, the solutions will often make use of TAP or SPAN ports to analyze the inline traffic stream's copy and will try to predict the attack based on a previously trained algorithm, hence making the need for a human intervention trivial[56].

Machine learning algorithms have been crucial in the world of cybersecurity. Especially, due to the incredible performance and potential of deep learning networks in recent days in various problems from a wide variety of fields which were considered unsolvable in the past, the reliability of applying it for Artificial Intelligence (AI) and unsupervised challenges have increased [39]. Deep learning is nothing but a partition of machine learning that mimics the functions of the human brain hence the name artificial neural network. The concept of deep learning consists of creating hierarchical representations that are complex and involve the creation of simple building blocks to solving high-level problems. Recently, deep learning techniques have been used in a variety of cyber security use cases, including [40,41,42,43,44,45, 46,47,48,49,50,51,52,53,54,55].

It follows that it is clear that when Deep Neural Networks and IDSs are joined, they can function at a superhuman level. Also, since the IDSs are out-of-band on the infrastructure, common attacks like DoS which primarily aim at choking the network band to gain access to the host, cannot bottleneck the performance of it, hence this security layer cannot tamper with ease. Towards The end, the sections are organised as follows: Section II reviews the work related to IDS, different deep neural networks and some discussions about the KDDCup-'99 dataset that was published. Section III takes an in-depth look at Deep Neural Networks (DNN) and applications of the ReLU activation function. Section IV analyses the dataset used in this paper, explains the shortcomings and often evaluates the final results. The conclusion of Section V outlines a likely workflow for this research project going forward.

2. RELATED WORK

Research on ID in network security has existed since the birth of computer architectures. The use of ML techniques and solutions to holistic IDS has become common in recent days, but training data at hand is limited and is mostly used only for benchmarking. One of the most comprehensive datasets that is publicly accessible is the DARPA dataset [1]. The data of the dump offered by the 1998 DARPAIDE valuation network 1998 was cleaned and utilized for the KDDCup-contest of 1999 at the 5th International Conference on Knowledge-edge Discovery and Data Mining. The job was to organise the records of the connections that are already preprocessed into either traffic which is normal, or one of the following categories of attack: 'DoS', 'Probing',' R2L'and'U2R'.

The MADAMID framework was used to preprocess the KDDCup-'99 competition's data [2]. The top three positions were taken by entries that used decision tree versions and showed only slight performance differences [3,4,5]. All 17 of the competition's initial submissions were evaluated and found to perform favourably [6]. The majority of published results were tested and trained with only a 10% training set observing the feature reduction on the KDDCup- '99' datasets [7, 8, 9]. Few researchers used custom-built datasets, extracted from the 10% KDDCup-'99' training set [10, 11,12].

Due to the usage of various training and test datasets, there are a lot of intriguing articles where the outcomes are inadvertently compared. In a paper [13], genetic algorithms and decision trees were used for automatic rule generation for an intelligent system for enhancing the capability of existing IDS. The integrated utilization of neural networks in IDS was suggested by [14] and [15]. [16] proposed an application of recurrent neural networks (RNNs) and [17] compared the neural network architectures' performance for statistical anomaly detection to datasets from four different scenarios.

Although the datasets of KDDCup-'99' have various issues [18], [19] argues that they are still an effective bench-marking dataset which is publicly available to compare different intrusion detection methods.

The fundamental reason for the popularity of ML-based approaches is because of their capability to attack the constantly evolving complex and diverse threats to achieve an acceptable false positive rate of ID with a reasonable computational cost. In the early stages, [36] used the PNrule method which is derived from P-rules and N-rules to figure out the existence and non-existence of the class respectively. This has an advantage due to the enhancement of the detection rate in the other types of attacks except for the U2R category.

An extrapolation to traditional Feed Forward Networks (FFN) in the plane of taking inspiration from biological elements, is a network named Convolutional Neural Network (CNN). In the early stages, CNN was used for processing images by making use of normal 2D layers, pooling 2D layers and completely connected layers. [37] studied the

applications of CNN for IDS with the KDDCup of the '99' dataset and compared the results with several other bleeding-edge algorithms. They have come to the broad conclusion that CNN is superior to the other algorithms. With the same dataset, [38] studied the application of the Long Short-Term Memory (LSTM) classifier. It has been stated that due of LSTM's capabilities to see into the past and relate the successive records of connections demonstrate usefulness towards intrusion detection systems.

The main goal of this study is to make use of the possibility of an inbound cyberattack's randomness, which is undetectable to the human eye but may be filtered by adding an artificial intelligence layer to the network. Hence, by training the neural network with the existing cyber attack data, one can learn to predict an inbound attack easily and can either alert the system or initiate a pre-programmed response which may prevent the attack from proceeding further. As a result, millions worth, of aftershock collateral damage and expensive data leaks can be prevented just by simply adding an extra layer to the security system. The benchmarking dataset used for training the networks is bygone and for a better real-time robustness of the algorithm, more recent data must be used for retraining before deploying in the field. The obligation of this paper is to introduce the essence of artificial neural networks in into the rapidly evolving field of cybersecurity.

3. BACKGROUND

Deep neural networks (DNNs) are Artificial Neural Networks (ANN) with a multi-layered structure comprised within the input-output layers. They can model complex non-linear relationships and can generate computational models where the object is expressed in terms of the layered composition primitives.

Below we roughly cover simple DNNs and applications of ReLU and why it is preferred over other activation functions.

1. Deep Neural Network(DNN)
2. Application of rectified linear units(ReLU)

3.1 Deep Neural Network(DNN)

While traditional machine learning algorithms are linear, deep neural networks are stacked in an increasing hierarchy of complexity as well as abstraction. Each layer applies nonlinear transformation onto its input and creates a statistical model as output from what it learns. In simple terms, the input layer is received by the input layer and passed onto the first hidden layer. These concealed layers analyse our inputs and make calculations. One of the challenges in creating neural networks is deciding the

hidden layers' count and the count of the neurons for each layer. An activation function is present in every neuron and is used to standardize the output of the neuron.

The "Deep" in deep learning refers to the presence of multiple hidden layers. The output layer returns the output data. Until the output has reached an acceptable level of accuracy, epochs are recontinued.

3.2 Application of rectified linear units(ReLU)

ReLU has turned out to be more efficient and has the capacity to accelerate the entire training process altogether [20].

Usually, Neural networks use a sigmoidal activation function or tanh (hyperbolic tangent) activation function. However, these functions are prone to vanishing gradient problems [21]. A vanishing gradient occurs when lower layers of aD have gradients of nearly null because units of higher layers, tanh function asymptotes are almost saturated. ReLU offers an alternative to sigmoidal nonlinearity which addresses the issues mentioned so far [22].

4. EXPERIMENTS

We consider Keras [23] as a wrapper on top of Tensor-Flow [24] as a software framework. A GPU-enabled TensorFlow in a single Nvidia-GK110BGL- Tesla-k40 has been used to significantly increase the agility of data processing in deep-learning systems.

4.1 Dataset Description

Lincoln Labs of MIT was in charge of organizing and managing the DARPA's initiative for ID evaluation in 1998. The main objective of this is to analyze and conduct research on ID. A standardized dataset was prepared, which included various types of intrusions which imitated a military environment and were made publicly available. The KDD intrusion detection contest's dataset of 1999 was a well-refined version of this[25].

4.2 Shortcomings of the KDDCup-'99 dataset

ReLu has turned out to be more efficient and has the detailed report and major shortcomings of the provided sin-thetic data set such as KDDCup-'98' and KDDCup-'99" were discussed by [26]. The main condemnation was that they failed to validate their data set a simulation of real-world profile. Irrespective of all these criticisms, the dataset KDD Cup-'99' has been used as an effective dataset by Byman researchers for benchmarking the IDS algorithms over the years. In contrast to the critiques about the creation of the dataset, [27] has revealed a detailed analysis of the contents, identified the non-uniformity and simulated the artefacts in simulated network trafficdata.

The reasons behind why the machine learning classifiers have limited capability in identifying the attacks that belong to the content categories R2L, and U2R in KDDCup-'99' data sets have been discussed by [28]. They have concluded that

it's not possible to get an acceptable detection rate using classical ML algorithms. It is also stated the possibility of getting a high detection rate in most of the cases by producing are fined and augmented data sets by combining the train and test sets. However, a significant approach has not been discussed.

The DARPA/KDDCup-'88 received significant criticism for failing to test the conventional IDS. To Eradicate this [29] used the Snort ID system on DARPA/ KDDCup-'98'tcpdumptraces. The system performed poorly resulting in low accuracy and impermissible false positive rates. It failed in detecting dos and probing category but contrasted performing better than the detection of R2L and U2R. Despite the harsh criticisms [30], the KDDCup-'99 season of the most widely used publicly available bench-marking datasets is reliable for studies related to IDS evaluation and other security-related tasks [31]. In the effort to mitigate the underlying problems existing with the KDDCup-'99' set, a fined version of the dataset named NSL-KDD was proposed by [31]. It removed the connection redundancy records of the entire train and test data. Additionally, the test data had the incorrect records eliminated. These measures prevent the classifier from being biased in the direction of more frequent records. Even after the refinement, this failed to solve the issues reported by [32, 33], and a new dataset named UNSW-NB15 was proposed.

4.2 DARPA/ KDDCup-'99'dataset

The DARPA's ID evaluation group, accumulated network-based data of IDS by simulation of an air force base over 1000s of UNIX nodes and for continuously 9 weeks,100s of users at a period of time in Lincoln Labs, which was then divided into training and testing periods of 7 and 2 weeks each, to extract the raw dump data TCP. MIT's lab with extensive support from DARPA and AFRL, used Windows and Linux nodes for almost all of the inbound intrusions from an alienated LAN, unlike other OS nodes. For the purpose of the dataset, 7 distinct scenarios and 32 distinct attacks total of up to 300 attacks which totals up to 300 attacks were simulated.

Since the year release of the KDD-'99 dataset [34], it's the most widely utilized data for evaluating several IDSs.This dataset is grouped together by almost 4,900,000 individual connections which includes a feature count of 41. The simulated attacks were roughly classed below.

4.2.1 Denial-of-Service-Attack (DoS)

Intrusion where a person aims to make a host inaccessible to its actual purpose by or sometimes permanently disrupting

services by flooding the target machine with enormous amounts of requests and hence overloading the host[35].

4.2.2 User-to-Root-Attack (U2R)

A category of commonly used manoeuvre by the perpetrator starts by trying to gain access to a user's pre-existing access and exploiting the holes to obtain root control.

4.2.3 Remote-to-Local-Attack (R2L)

The intrusion in which the attacker can send data packets to the target but has no user account on that machine itself, tries to exploit one vulnerability to acquire local access by posing as the target machine's active user to do so.

4.2.4 Probing-Attack

The type in which the perpetrator tries to gather information about the computers of the network and the ultimate aim for doing so is to get past the firewall and gaining root access.

4.2.5 Groups

KDDCup-'99' set is classified into the following three groups: Basic features: Attributes obtained from a connection of TCP/IP come from this group. The majority of these features result in implicitly delaying the detection. Featured in traffic Features This group includes calculations made in relation to a specific window of time. This can be broken into two groups further:

(i) **"Same host"** features: Connections that are used to calculate statistics on protocol behavior, etc. are those that keep the initial end host as the connection under consideration for a continuous 2 seconds.

(ii) **"Same Service"**: Features that fall under this category include connections that have only provided the same services as the current connection for the past two seconds.

4.2.6 Content features

Generally probing attacks and DDoS attacks have at least some kind of frequent sequential intrusion patterns, unlike R2L and U2R attacks. This is due to the reason that they involve multiple connections to a single set of a host(s) under a short span of time while the other 2 intrusions are integrated into the packets of data partitions in which generally only one connection is involved. We require some distinct features for the detection of these attacks so that we can look for some unusual behavior. These are called content features.

4.3 Identifying Network Parameters

Hyper-tuning of parameters to figure out the optimum set of parameters to achieve the desired result is all by itself separate field with plenty of future scope for research. In this paper, the learning is kept constant at 0.01 while the other parameters are optimized. The count of the neurons in the layer was experimented with by changing it over the range of 2 to 1024. The count was then raised to 1280, however, that didn't result in a noticeably higher degree of precision. Therefore, the neuron count was tuned to 1024.

4.4 Identifying Network Structures

Conventionally, increasing the count of the layers results in better results compared to increasing the neuron count in a layer. In order to examine and determine the best network structure for our input data, the following network topologies were examined.

4.4.1 1, 2, 3, or 4 Layer DNN

100 epochs were conducted for each of the aforementioned network topologies, and the results were shown. Finally, the best performance was shown by the DNN 3 layer compared to all the others. To broaden the search for better results, all the common classical machine learning algorithms were used and the results were compared to the DNN 3 layer, which still outperformed every single classical algorithm. Table I shows the comprehensive statistical findings for various network architectures.

4.5 Proposed Architecture

Fig. 1 depicts an overview of the proposed DNN architecture for all use cases. This comprises a hidden-layer count of 5 and an output-layer. The input layer consists of 41 neurons. The Neurons Input-layer hidden-layer hidden to output-layer are connected completely. Back-propagation mechanism is used to train the DNN networks. To make the network more robust, the suggested network consists of fully linked layers, bias layers, and dropout layers.

Table -1: Network Structure Information

Layer (type)	Output Shape	Param
Dense–1 (Dense)	(NIL, 1024)	43008
Dropout-1 (Dropout)	(NIL, 1024)	0
Dense–2 (Dense)	(NIL, 768)	787200
Dropout-2 (Dropout)	(NIL, 768)	0
Dense–3 (Dense)	(NIL, 512)	393728
Dropout-3 (Dropout)	(NIL, 512)	0
Dense–4 (Dense)	(NIL, 256)	131328
Dropout-4 (Dropout)	(NIL, 256)	0
Dense–5 (Dense)	(NIL, 128)	32896
Dropout-5 (Dropout)	(NIL, 128)	0
Dense–6 (Dense)	(NIL, 1)	129
Activation-1 (Activation)	(NIL, 1)	0

This layer's input and hidden layers are made up of 41 neurons. These are then fed into the hidden layers. ReLU serves as the non-linear activation function for hidden layers. Then weights are added to feed them forward to the next hidden layer. The neuron count in each hidden layer is decreased steadily from the first to the output to make the outputs more accurate and at the same time reduce the computational cost.

Regularization: To speed up and save time on the entire procedure, Dropout (0.01). The dropout's purpose is to randomly disconnect the neurons, which strengthens the model and keeps it from over-fitting the training data.

Layer and classification of output: There are only two neurons in the outer layer. Benign and offensive. Since the 1024 neurons from the previous layer must be converted into just 2 neurons, a sigmoid activation function is used. Due to the nature of the sigmoid function, it returns only two outputs, hence favouring the binary classification that was intended in this paper.

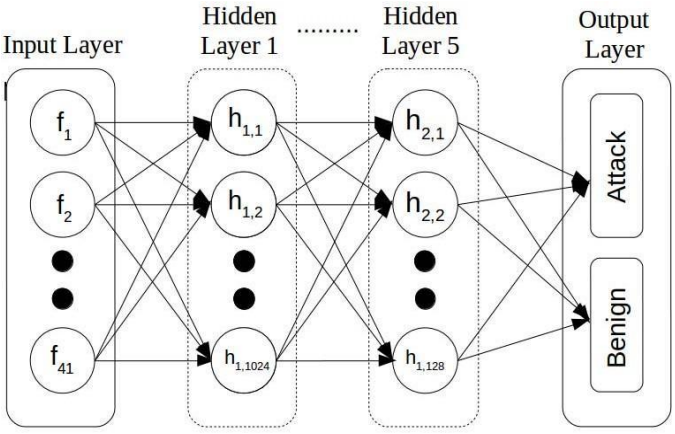

Fig -1: Proposed architecture

5. RESULTS

The KDDCup-'99 dataset was fed through traditional machine learning algorithms as well as DNNs with different hidden layers for the purposes of this work.

Table -2: Result

Algorithm	Accuracy	Precision	Recall	F1-Score
DNN-1	0.929	0.998	0.915	0.954
DNN-2	0.929	0.998	0.914	0.954
DNN-3	0.930	0.997	0.915	0.955
DNN-4	0.929	0.999	0.913	0.954
DNN-5	0.927	0.998	0.911	0.953
Ada Boost	0.925	0.995	0.911	0.951
Decision Tree	0.928	0.999	0.912	0.953
K-Nearest Neighbor	0.929	0.998	0.913	0.954
Linear Regression	0.848	0.989	0.821	
Navie Bayes	0.929	0.988	0.923	0.955
Random Forest	0.927	0.999	0.910	0.953
SVM*-Linear	0.811	0.994	0.770	0.868
SVM*-rbf	0.811	0.992	0.772	0.868

(* Support Vendor Machine)

After the training was completed, all models were compared for f1-score, accuracy, recall and precision with the test dataset. The scores for the same have been compared in detail in Table -2.

DNN 3-layer network has outperformed all the other classical machine learning algorithms. It is so because of the ability of DNNs to extract data and features with higher abstraction and the non-linearity of the networks adds up to the advantage when compared with the other algorithms.

6. FUTURE SCOPE

For better efficiency, our system must be real-time processing which should be achieved in future. In the future alerts generated by IDS are sent to the Oracle database an actual email is generated and information about the administrator and local user is added to the email.

7. CONCLUSION

This paper has comprehensively recapitulated the usefulness of DNs in IDS comprehensively. For the purpose of reference, other classical ML algorithms have been accounted for and compared against the results of DNN. The publicly available KDDCup-'99 dataset has been primarily used as the benchmarking tool for the study, through which the superiority of the DNN over the other compared algorithms has been documented clearly. For further refinement of the algorithm, this paper takes into account DNNs with different counts of hidden layers and it was concluded that a DNN with 3 layers has been proven to be effective and accurate of all.

An issue with this process is that the neurons are trained using an outdated benchmarking dataset, as this paper has discussed numerous times. Fortunately, it can be vanquished by using a fresh dataset with the essence of the latest attack strategies before the actual deployment of this artificial intelligence layer to the current network infrastructures to guarantee the adaptability of the algorithm in the real world.

From the empirical results of this paper, we may claim that deep learning methods are a promising direction towards cyber security tasks, but even though the performance on

artificial dataset is exceptional, application of the same on network traffic in the real-time which contains more complex and recent attack types is necessary. Additionally, studies regarding the flexibility of these DNNs in adversarial environments are required. The increase in vast variants of deep learning algorithms calls for an overall evaluation of these algorithms in regard to their effectiveness towards IDSs. This will be one of the directions IDS research can travel and hence will remain as a work of the future.

REFERENCES

[1] R. Lippmann, J. Haines, D. Fried, J. Korba and K. Das. "The 1999 DARPA off-line intrusion detection evaluation". Computer networks, vol. 34, no. 4, pp. 579– 595, 2000. DOI http://dx.doi.org/10.1016/S1389-1286(00)00139–0.

[2] W. Lee and S. Stolfo. "A framework for constructing features and models for intrusion detection systems". ACM transactions on information and system security, vol. 3, no. 4, pp. 227–261, 2000. DOI http://dx.doi. Org/10.1145/382912.382914.

[3] B. Pfahringer. "Winning the KDD99 classification cup: Bagged boosting". SIGKDD explorations newsletter, vol. 1, pp. 65–66, 2000. DOI http://dx.doi.org/10. 1145/846183.846200.

[4] M. Vladimir, V. Alexei and S. Ivan. "The MP13 approach to the KDD'99 classifier learning contest". SIGKDD explorations newsletter, vol. 1, pp. 76– 77, 2000. DOI http://dx.doi.org/10.1145/846183. 846202.

[5] R. Agarwal and M. Joshi. "PNrule: A new framework for learning classier models in data mining". Tech. Rep. 00–015, Department of Computer Science, University of Minnesota, 2000.

[6] C. Elkan. "Results of the KDD'99 classifier learning". SIGKDD explorations newsletter, vol. 1, pp. 63– 64, 2000. DOI http://dx.doi.org/10.1145/846183. 846199.

[7] S. Sung, A.H. Mukkamala. "Identifying important features for intrusion detection using support vector machines and neural networks". In Proceedings of the symposium on applications and the Internet (SAINT), pp. 209–216. IEEE Computer Society, 2003. DOI http: //dx.doi.org/10.1109/saint.2003.1183050.

[8] H. Kayacik, A. Zincir-Heywood and M. Heywood. "Selecting features for intrusion detection: A feature relevance analysis on KDD 99 intrusion detection datasets". In Proceedings of the third annual conference on privacy, security and trust (PST). 2005.

[9] C. Lee, S. Shin and J. Chung. "Network intrusion detection through genetic feature selection". In Seventh ACIS international conference on software engineering, artificial intelligence, networking, and parallel/distributed computing (SNPD), pp. 109–114. IEEE Computer Society, 2006

[10] S. Chavan, K. Shah, N. Dave, S. Mukherjee, A. Abraham and S. Sanyal. "Adaptive neuro-fuzzy intrusion detection systems". In Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC), vol. 1, pp. 70–74. IEEE Computer Society, 2004. DOI http://dx.doi.org/ 10.1109/itcc.2004.1286428.

[11] S. Chebrolu, A. Abraham and J. Thomas. "Feature deduction and ensemble design of intrusion detection systems". Computers \& security, vol. 24, no. 4, pp. 295–307, 2005. DOI http://dx.doi.org/10.1016/j. Cose.2004.09.008.

[12] Y. Chen, A. Abraham and J. Yang. "Feature selection and intrusion detection using hybrid flexible neural tree". In Advances in Neural Networks (ISNN), vol. 3498 of Lecture notes in computer science, pp. 439– 444. Springer Berlin / Heidelberg, 2005. DOI http://dx.doi.org/10.1007/11427469_71.

[13] C. Sinclair, L. Pierce and S. Matzner. "An application of machine learning to network intrusion detection". In Proceedings of the 15th annual Computer Security Applications Conference (ACSAC), pp. 371–377. IEEE Computer Society, 1999. DOI http://dx.doi.org/ 10.1109/csac.1999.816048.

[14] H. Debar, M. Becker and D. Siboni. "A neural network component for an intrusion detection system". In Proceedings of the IEEE Computer Society Symposium on research in security and privacy, pp. 240–250. IEEE Computer Society, 1992. DOI http://dx.doi.org/ 10.1109/risp.1992.213257.

[15] J. Cannady. "Artificial neural networks for misuse detection". In Proceedings of the 1998 National Information Systems Security Conference (NISSC), pp. 443–456. Citeseer, 1998.

[16] H. Debar and B. Dorizzi. "An application of a recurrent network to an intrusion detection system". In an International joint conference on neural networks, 1992. IJCNN., vol. 2, pp. 478 –483 vol.2. jun 1992. DOI http://dx.doi.org/10.1109/ijcnn. 1992.226942.

[17] Z. Zhang, J. Lee, C. Manikopoulos, J. Jorgenson and J. Ucles. "Neural networks in statistical anomaly intrusion detection". Neural network world, vol. 11, no. 3, pp. 305–316, 2001.

[18] J. McHugh. "Testing intrusion detection systems: A critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln

Laboratory". ACM transactions on information and system security, vol. 3, no. 4, pp. 262–294, 2000. DOI http://dx.doi.org/10.1145/382912.382923.

[19] M. Tavallaee, E. Bagheri, W. Lu and A. A. Ghorbani. "A detailed analysis of the KDD CUP 99 data set". In IEEE symposium on computational intelligence for security and defence applications, Cisda, pp. 1–6. IEEE, Jul. 2009. DOI http://dx.doi.org/10.1109/cisda. 2009.5356528.

[20] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, 2011, pp. 315–323.

[21] Bengio, Y., Simard, P. and Frasconi, P., 1994. Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks, 5(2), pp.157–166.

[22] Maas, A.L., Hannun, A.Y. and Ng, A.Y., 2013, June. Rectifier nonlinearities improve neural network acoustic models. In Proc. icml (Vol. 30, №1, p. 3).

[23] F. Chollet, "Keras (2015)," URL http://keras. Io, 2017.

[24] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard et al., "Tensorflow: A system for large-scale machine learning." in OSDI, vol. 16, 2016, pp. 265–283.

[25] Stolfo, S., Fan, W. and Lee, W., KDD-CUP-99 Task Description. 1999–10–28)[2009–05–08]. http://KDD. ics. uci. edu/databases/kddcup99/task, html.

[26] J. McHugh. "Testing intrusion detection systems: A critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory". ACM transactions on information and system security, vol. 3, no. 4, pp. 262–294, 2000. DOI http://dx.doi.org/10.1145/382912.382923.

[27] M. Mahoney and P. Chan. "An analysis of the 1999 DARPA/Lincoln Laboratory evaluation data for network anomaly detection". In Recent advances in intrusion detection, vol. 2820 of Lecture notes in computer science, pp. 220–237. Springer Berlin / Heidelberg, 2003.

[28] Sabhnani, Maheshkumar, and Gursel Serpen. "Why machine learning algorithms fail in misuse detection on KDD intrusion detection data set." Intelligent Data Analysis 8, no. 4 (2004): 403–415.

[29] S. Brugger and J. Chow. "An assessment of the DARPA IDS evaluation dataset using snort". Tech. Rep. CSE-2007–1, Department of Computer Science, University of California, Davis (UCDAVIS), 2005.

[30] J. McHugh. "Testing intrusion detection systems: A critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory". ACM transactions on information and system security, vol. 3, no. 4, pp. 262–294, 2000. DOI http://dx.doi.org/10.1145/382912.382923.

[31] Tavallaee, Mahbod, Ebrahim Bagheri, Wei Lu, and Ali-A. Ghorbani. "A detailed analysis of the KDD CUP 99 data set." In Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications 2009. 2009.

[32] Moustafa, Nour, and Jill Slay. "The evaluation of Network Anomaly Detection Systems: Statistical analysis of the U[8] H. Kayacik, A. Zincir-Heywood and M. Heywood. "Selecting features for intrusion detection: A feature relevance analysis on KDD 99 intrusion detection datasets". In Proceedings of the third annual conference on privacy, security and trust (PST). 2005.

[33] Moustafa, Nour, and Jill Slay. "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data)."Military Communications and Information Systems Conference (MilCIS), 2015. IEEE, 2015.

[34] KDD Cup 1999. Available on: http://kdd.ics.uci.edu/database[8] H. Kayacik, A. Zincir-Heywood and M. Heywood. "Selecting features for intrusion detection: A feature relevance analysis on KDD 99 intrusion detection datasets". In Proceedings of the third annual conference on privacy, security and trust (PST). 2005.

[35] McDowell, M. (2013). Understanding Denial-of-Service Attacks US-CERT. United States Computer Emergency Readiness Team.

[36] R. Agarwal and M. V. Joshi, "Pnrule: A new framework for learning classifier models in data mining (a case study in network intrusion detection)," in Proceedings of the 2001 SIAM International Conference on Data Mining. SIAM, 2001, pp. 1–17.

[37] Vinayakumar, R., Soman, K. P., \& Poornachandran, P. (2017, September). Applying convolutional neural network for network intrusion detection. In Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on (pp. 1222–1228). IEEE.

[38] Staudemeyer, R. C. (2015). Applying long short-term memory recurrent neural networks to intrusion detection. South African Computer Journal, 56(1), 136–154.

[39] Sommer, R. and Paxson, V., 2010, May. Outside the closed world: On using machine learning for network intrusion detection. In Security and Privacy (SP), 2010 IEEE Symposium on (pp. 305–316). IEEE.

[40] Vinayakumar, R., Soman, K. P., \& Poornachandran, P. (2018). Evaluating deep learning approaches to characterize and classify malicious URLs. Journal of Intelligent \& Fuzzy Systems, 34(3), 1333–1343.

[41] Vinayakumar, R., Soman, K. P., \& Poornachandran, P. (2018). Detecting malicious domain names using deep learning approaches at scale. Journal of Intelligent \& Fuzzy Systems, 34(3), 1355–1367.

[42] Vinayakumar, R., Soman, K. P., Poornachandran, P., \& Sachin Kumar, S. (2018). Evaluating deep learning approaches to characterize and classify the DGAs at scale. Journal of Intelligent \& Fuzzy Systems, 34(3), 1265–1276.

[43] Vinayakumar, R., Poornachandran, P., \& Soman, K. P. (2018). Scalable Framework for Cyber Threat Situational Awareness Based on Domain Name Systems Data Analysis. In Big Data in Engineering Applications (pp. 113–142). Springer, Singapore.

[44] Vinayakumar, R., Soman, K. P., Poornachandran, P., \& Sachin Kumar, S. (2018). Detecting Android malware using long short-term memory (LSTM). Journal of Intelligent \& Fuzzy Systems, 34(3), 1277–1288.

[45] Vinayakumar, R., Soman, K. P., \& Poornachandran, P. (2017, September). Deep encrypted text categorization. In Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on (pp. 364–370). IEEE.

[46] Vinayakumar, R., Soman, K. P., \& Poornachandran, P. (2017, September). Secure shell (SSH) traffic analysis with flow-based features using shallow and deep networks. In Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on (pp. 2026–2032). IEEE.

[47] Vinayakumar, R., Soman, K. P., \& Poornachandran, P. (2017, September). Evaluating shallow and deep networks for secure shell (SSH) traffic analysis. In Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on (pp. 266–274). IEEE.

[48] Vinayakumar, R., Soman, K. P., Velan, K. S., \& Ganorkar, S. (2017, September). Evaluating shallow and deep networks for ransomware detection and classification. In Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on (pp. 259–265). IEEE.

[49] Vinayakumar, R., Soman, K. P., \& Poornachandran, P. (2017, September). Applying deep learning approaches for network traffic prediction. In Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on (pp. 2353–2358). IEEE.

[50] Vinayakumar, R., Soman, K. P., \& Poornachandran, P. (2017, September). Long short-term memory-based operation log anomaly detection. In Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on (pp. 236–242). IEEE.

[51] Vinayakumar, R., Soman, K. P., \& Poornachandran, P. (2017, September). Deep android malware detection and classification. In Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on (pp. 1677–1683). IEEE.

[52] Vinayakumar, R., Soman, K. P., \& Poornachandran, P. (2017, September). Applying convolutional neural network for network intrusion detection. In Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on (pp. 1222–1228). IEEE.

[53] Vinayakumar, R., Soman, K. P., \& Poornachandran, P. (2017, September). Evaluating the effectiveness of shallow and deep networks to the intrusion detection system. In Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on (pp. 1282–1289). IEEE.

[54] Vinayakumar, R., Soman, K. P., \& Poornachandran, P. (2017). Evaluation of Recurrent Neural Network and its Variants for Intrusion Detection System (IDS). International Journal of Information System Modeling and Design (IJISMD), 8(3), 43–63.

[55] Vysakh S Mohan, Vinayakumar R, Soman Kp and Prabaharan Poornachandran. S.P.O.O.F Net: Syntactic Patterns for identification of Ominous Online Factors In Security and Privacy Workshops (SPW), 2018 IEEE [InPress].

[56] LeCun, Y., Bengio, Y. and Hinton, G., 2015. Deep learning. nature, 521(7553), p.436.