IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Issue-4, April 2019 E-ISSN: 2347-2693

Bi-Directional Recurrent Neural Network for IDS in the Internet of Things

Susheel Kumar Tiwari[1*], Manmohan Singh[2], Rahul Sharma[3]

[1]Department of Computer Science Engineering, Millennium Institute of Technology & Science, Bhopal
[2]Department of Computer Science Engineering, Chameli Devi Group of Institution, Indore
[3]Department of Computer Science Engineering, Chameli Devi Group of Institution, Indore

**Corresponding Author: sushiltiwari24@yahoo.co.in*

DOI: https://doi.org/10.26438/ijcse/v7i4.12271235 | Available online at: www.ijcseonline.org

Accepted: 20/Apr/2019, Published: 30/Apr/2019

Abstract— With IoT bringing a large number of day-by-day objects into the digital fold to make them smarter. It is also evident that the IoT is going to transform into a multi-trillion-dollar industry in the near future. However, the reality is that IoT bandwagon rushing full steam ahead is prone to count-less cyber- attack's in the extremely hostile environment like the internet. Nowadays, standard PC security solutions won't solve the challenge of privacy and data security transmitted over the internet. In this Paper, we have applied a Bidirectional Recurrent Neural Network to build a security solution with high durability for IoT network security. DL and ML have shown remarkable result in dealing with multimodal and voluminous hetero-generous data in regard's to intrusion detection especially with the architectures of Recurrent Neural Network's. Feature selection mechanism were also implemented to help identify and remove non-essential variables from data that does not affect the accuracy of the prediction model. In this case a Random Forest algorithm was implemented over Principal Component Analysis because of flexibility, and easy in using machine learning algorithms that allow production without hyper-parameter tuning, building of multiple decision tree and merging them together to get a more accurate and stable prediction. In this study a novel algorithm (BRNN) out-performed both Recurrent Neural Network and Gated Recurrent Neural Network because it consider both information from the past and the future with back and forward hidden neuron's.

Keywords— IoT, Recurrent Neural Network's, Bi-Directional RNN, Intrusion Detection, Deep Learning, Machine Learning.

I. INTRODUCTION

Internet of Things is an upcoming technology that transforms everyday physical objects into an eco-system that can enrich and simplify our lives by influencing human routine toward's, e-health, e-learning, remote monitoring, surveillance [1][2].

IoT plays a key role in industries such as automations and intelligent industrial manufacturing, smart logistics, smart transportation and so forth. IoT technology is bringing a large number of day-to-day object into the digital fold to make them smarter. It is also evident that IoT is going to transform into a Multi-Trillion-Dollar industries in the near future. It is expected that until 2022[3] we will have around 50 billion devices connected to the network, which is a 140 percent increase compared to 2018. And in 2035, this number could reach 1 trillion devices' [4], with the IoT bandwagon rushing full steam ahead. There are enormous security risk's associated with the device's. The influx of additional entry point into an organization network, plus a current lack of security standard for IoT devices, means there is a gaping hole in the perimeter of any home or business that has installed IoT devices. The crosscutting nature of IoT systems and the multidisciplinary components involved in the deployment of such systems introduced new security challenges. To tackle those issue with IoT complexities, we could use the concepts of "lightweight" and "adoption" to develop robust security solutions. "Adaptive Lightweight" solutions have proven their worth multiple times in dealing with inconsistencies in very large distributed systems. It is almost impossible to design security solution for each IoT device in a network because their large number. However, secure data in transit to and from the connection between the device's in an IoT network would be a practical approach. With the help of DL algorithm BRNN, a range of sizes and types of data can be analyzed to develop adaptive solutions for the IoT system.

The biggest benefit that DL algorithm brings to IoT is the automation analysis of colossal amounts of generated and exchanged data. Instead of a human data analyst going through all these data manually, looking for pattern and anomalies, with properly implemented DL algorithm we can use a completely reversed top-down approach in analysis.

This research study the effectiveness Bi-directional recurrent neural network for intrusion detection which has

provided promising results compared to some literature work. The full KDD - Cup-99 Intrusion Detection dataset [5, 32] were used to evaluate the algorithm.

1.1 Motivation

DL is an AI function that imitates the workings of the human brain in processing data and creating patterns for use in decision making. All of the connected sensor's that make up the IoT are like our bodies, they provide the raw data of what is going on in the world [6].

Artificial intelligence (AI) is like our brain, making sense of that data and deciding what actions to perform. And the connected devices of Internet of Things (IoT) are again like our bodies, carrying out physical actions or communicating with others [7].

BRNN ability to predict both the positive and negative directions of time simultaneously allowing them to receive information from both past and future states. Based on that architectures with high computation power we believe that it has great potential to find more insights from IoT network data traffic. Despite BRNN architecture being complex, with hyper-parameters tuning Internet of Things (IoT) security solution can be efficiently obtained. This served as motivation to apply BRNN for intrusion detection.

1.2 Problem Statement.

One of the long-lasting problems with Internet of things (IoT) is that much of the transmitted information is not adequately secured. IoT devices are connected for longer time periods without human intervention and network threats are evolving at an unprecedented rate. Now-a-days, standard PC security solutions will not solve the challenge for the fact that IoT is dealing with hetero-geneous data of various sizes in multi-modal systems [14].

The aim of this research paper is to analyze and answer the following research questions:

- What are the security issues for the Internet of Things
- Does Bi-Directional Recurrent Neural Network outperform other machine learning approaches for Intrusion Detection classification on the IoT?
- What is the set of hyper parameters that hel ps to achieve high accuracy and less time in training?

1.3 Current solution

The current security protocols of securing traditional PCs, servers and mobile devices for detecting anomalies, is only applicable for high powered computers for short -lived session. Hence, they are not strong enough for network threats evolving at an unprecedented rate. It is not viable to use the same protection technique for long-running session. For these reasons, Internet of Things devices became attractive targets for hackers making people's live endangered with unexpected threats. Traditional systems were designed to find better- known attacks, but they cannot determine unknown threats.

II. BACKGROUND

2.1 What is the IoT?

The Internet of Things (IoT) is a computing concept that describes the idea of everyday physical objects being connected to the internet and being able to identify themselves to other devices [9, 10]. The Internet of Things (IoT) describes the revolution already under way that is seeing a growing number of internets enabled devices that can network and communicate with each other and with other web-enabled gadgets [11]. It enriches our lives and make it simpler by making easy and possible machine-to-machine communication and machine-to human communication. With such wide offerings and futuristic scope real-world use cases of internet of things in this context are smart grids, smart homes, smart cities and the Industrial Internet of Things (IIoT).

Because IoT covers a distinct number of protocols, domains, and applications. There will be more advanced communication between the devices with better connectivity and services.

2.2 Privacy and security issues in IoT.

Whatever the future brings you must not lose sight; the Internet of Things is likely to meld the virtual and physical worlds together in ways that are currently difficult to comprehend. But from a security and privacy perspective this raise a serious challenges.

The smart, connected objects that will densely populate the Internet of Things will interact with both human's and the human environment by providing, processing, and delivering all sorts of information and command's. These connected things will be able to communicate information about individuals and objects, their state, and their surroundings, and can be used remotely. All of this connectivity carries with it a risk to privacy and information leakage. [12]

The IoT raises is vast in terms of infrastructures, network, device, and interface.

2.3 Intrusion Detection System (IDS).

Intrusion detection describe's an application security practice used to mitigate attacks and block new threats. It is a reactive measure that identifies and mitigates ongoing attacks using an intrusion detection system. It's able to weed out existing malware (e.g., Trojans, backdoors, root kits) and detect social engineering assaults that manipulate users into revealing sensitive information [40]. Upon detecting a security policy violation, virus or configuration error, an Intrusion Detection System is able to kick an offending user off the network and send an alert to security personnel.

Despite its benefits, including in-depth network traffic analysis and attack detection, an IDS has inherent drawbacks. Because it uses previously known intrusion signatures to locate attacks, newly discovered (i.e., zero-day) threats can remain undetected.

Based on their responsive nature, IDS is categorized into Active IDS and Passive IDS. An Active IDS is designed to block the malware attacks automatically, without any human intervention, whereas a passive IDS only monitors the network traffic and alerts the users. Another categorization of IDS is Signature-Based IDS and Anomaly-based IDS. In the signature-based approach, the IDS access a database of known signatures and vulnerabilities. The simulated attacks fall in one of the following four categories:

Denial of Service Attack (DoS): is an attack in which the attacker makes some computing or memory resource too busy or too full to handle legitimate requests, or denies legitimate users access to a machine.

User to Root Attack: is a class of exploit in which the attacker starts out with access to a normal user account on the system (perhaps gained by sniffing passwords, a dictionary attack, or social engineering) and is able to exploit some vulnerability to gain root access to the system.

Remote to Local Attack: occurs when an attacker who has the ability to send packets to a machine over a network but who does not have an account on that machine exploits some vulnerability to gain local access as a user of that machine.

Probing Attack: is an attempt to gather information about a network of computers for the apparent purpose of circumventing its security controls.

III. RELATED WORK

Several researchers have applied deep learning and machine learning algorithm successfully for detecting intrusions and the results are showing great improvements compared to conventional methods.

Recent work by Roy [41] exploited the BRR algorithm to investigate and explains the efficiency of DL algorithms towards intrusion detection in Internet of Thing systems. The algorithm was trained on UNSW-NB14 and has achieved the accuracy of 97.00% classifying intrusion as normal or attack also, W. Anani [42] evaluated the performance of RNNS on full KDD cup intrusion detection system and the results show that vanilla LSTM recorded the best accuracy of 99.48% compared to the enhanced version of LSTM, dynamic RNNs recorded the best accuracy performance but took more time to train.

H. Hindy, E. Hodo, E .Bayne et al presented a neural network- based approach for intrusion detection on IoT network to identify DDoS/DOS attacks. The detection was based on classifying normal and threat patterns. The ANN model was validated against a simulated IoT network demonstrating over 99%accuracy. It successfully identified different types of attacks and showed good results for true and false positive rates performance.

C. Yin, Y. Zhu, J. Fei et al. (2017)[44] explore the RNN-IDS for both binary and multiclass classification with 97.09% of accuracy the algorithm was evaluated on NSL-KDD dataset using the fully connected model and proven that it has stronger modeling ability and higher detection rate than the reduced-size RNN model.

CONGYUAN XU applied GRUs combined with MLP to identify network intrusion for both NSL-KDD and KDD dataset and the model achieved 99.42% on KDD99 and 99.31% on NSL-KDD, with false-positive rates as low as 0.05% and 0.84%, respectively. particularly [45], the detection rates for DOS attacks were 99.98% on KDD 99 and99.55% on NSL- KDD.

The Long-Short-Term-Memory algorithm along with Gradient Descent Optimization was used by Kim. J to classify intrusion detection and the results recorded were promising with a precision of 97.54% and recall of 98.95% [47], they also introduced Gated Recurrent Unit for the first time in the research on intrusion detection data sets and the results obtained for recall, false alarm rate and accuracy are 97.06%, 10.01% and 98.65% [48]. Also, Staudemeyer, R. C., (2013, 23 October) evaluated the LSTM network's performance on the KDD 99 'Cup IDS data set but his results were improved and the results for cost training the network and network accuracy 22.13 and 93.82% [46].

3.1 Evaluation metrics

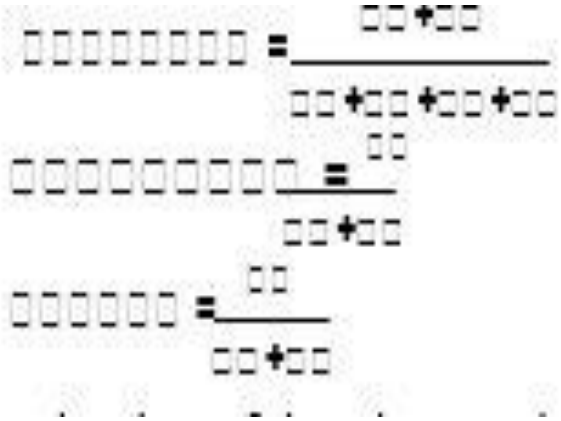
To evaluate the performance of the classification, model the following metrics are used in machine learning research. In general, the confusion matrix visualizes the performance of the algorithm in a tabular form as shown in the figure below:

Table1: Depicting table for evaluation metrics

	Predicted as Normal	Predicted as Attack
Actual Normal	TP	FP
Actual Attack	FN	TN

1. True Positives (TP): when the actual class of the data point was 1(True) and the predicted is also 1(True)
2. True Negatives (TN): when the actual class of the data point was 0(False) and the predicted is also 0 (False)
3. False Positives (FP): when the actual class of the data point was 0(False) and the predicted is 1(True).
4. False Negatives (FN): When the actual class of the data point was 1(True) and the predicted is 0 (False).

From the above table which actually represent the confusion matrix other important metrics such as Precision, Accuracy, Recall, False Alarm Rate (FAR) can be calculated:



Accuracy: The ratio between the class of data that are classified correctly and the total data (out of all the data , how many are correctly classified).
Precision: Out of all data that are predicted to be positive, how many are actually positive?
Recall: Out of all positive data, how many are actually positive?

3.2 Random forest classifier

Random Forest is a flexible, easy to use machine learning algorithm that produces great results most of the time, even without hyper-parameter tuning. It is also one of the most used algorithms because of its simplicity and the fact that it can be used for both classification and regression tasks. Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction [[34, 39]

We have made use of the ability of the random classifier method to rank the importance of the features set to the target variables. We have selected those variables based on the maximum importance levels. Those features with low values of the importance will add less information to the learning model and are ignored based on the threshold values of the importance.

3.3 IDS - datasets:

Intrusion detection and anomaly detection Researchers mainly use two datasets which are the UNB ISCX 2012 datasets and KDD Cup'99 /DARPA datasets. The most literature for the evaluation of anomaly detection methods uses the DARPA KDD Cup '99 dataset and by selecting this for our research it allows us to compare the results obtained to the results of the previous research. The DARPA KDD Cup '99 datasets were generated by the Defense Advanced Research Projects Agency (DARPA ITO) on a simulated air force model [4][19].

The 10.00% KDD dataset was selected which contains 24 attack types, which are mainly categorized into four classes – Probe, Denial of Service(DoS), User to Root (U2R) and Remote to Local (R2L).[31] The training and testing samples are Represented with 41 features and a label with either "normal" or "attack type".

IV. PROPOSED METHODOLOGY

4.1 Bi-directional RNNS:

Bidirectional RNNs are based on the idea that the output at time t may depend on previous and future elements in the sequence. To realize this, the output of two RNN must be mixed: one to executes the process in a direction and the second runs the process in the opposite direction. The network splits neurons of a regular RNN into two directions, one for positive time direction (forward states), and another for negative time direction (backward state's). By this structure, the output layer can get information from past and future states[18].and this overcomes the gap missing from Gated recurrent neural network and RNN because the RNN model has a major drawback called the vanishing gradient problem. The vanishing gradient problem means that since at each time-step during training the same weights is used to calculate the output. Also, it is hard to remember values from long way in the past for them, hence the result might not be accurate.

GRNN introduced by Cho, et al. in 2014, GRU (Gated Recurrent Unit) aiming to solve the vanishing gradient problem with GRU uses the so-called, update gate and reset gate which allows a GRU to carry forward information over many time periods in order to influence a future time period[28]. Moreover, if it was implemented it will not achieve higher accuracy compared to BRNN because it has less cell compared to the BRNN. Google's TensorFlow core

1.10.0 was used to perform the experiments as it provides an option to visualize the network design which is important for the developers. The following were used to perform necessary experiments:

Programming Language: Python3.6.5, Libraries used: NumPy1.14.3, scikit-learn 0.19.1, pandas 0.23.0, and TensorFlow. 1.10.0.

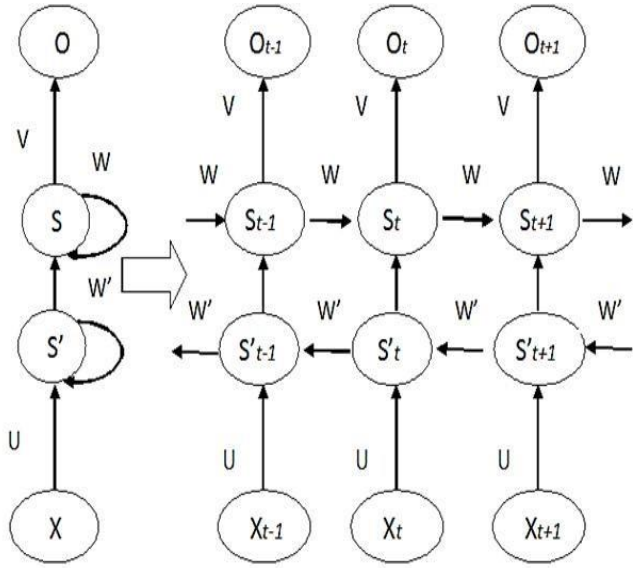


Fig.1: The unrolled architecture of Bi-RNN.

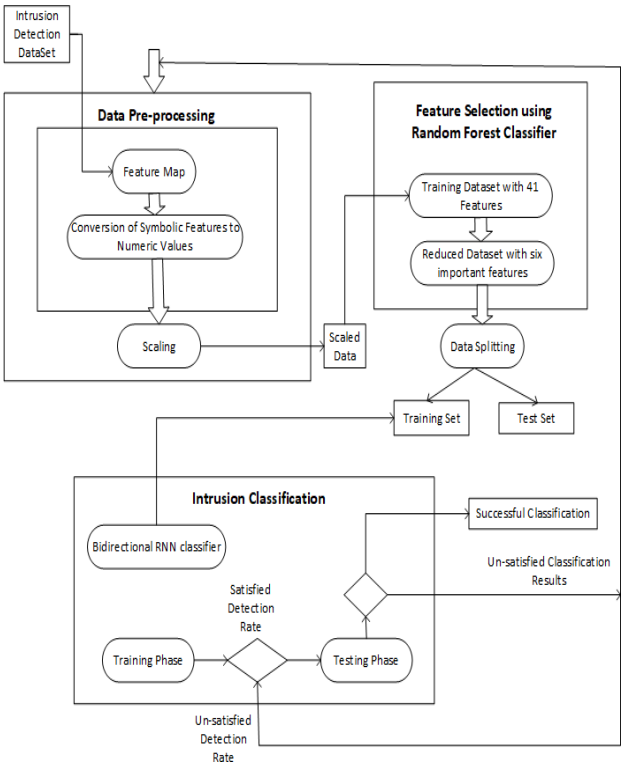


Fig.2: End to -end data flow of our deep learning model

V. RESULTS, ANALYSIS, AND DISCUSSION

In this part results are discussed in detail for each IDS classifier obtained using Bi-directional recurrent neural network(BRNN) and their evaluation measures.

Based on the architecture we performed sets of experiments using different hyperparameters(learning rate, time-steps, hidden layer).To optimize the results we tuned the Hyper-parameters. Since this is a binary classification it was classified as normal or attack for each sample and the best model was decided by considering every relevant metric.

5.1 Feature Selection:

Random forest classifier algorithm was used to select the important features which are relevant to the model and all the results are shown in table 2 and figures 10,11,12,13.

Table 2: Selected Features list for each IDS classifier based on the performance

Layer Type	Features Selected
All layers	Protocol type, service, flag, src_bytes, dst_bytes, logged_in, count,srv_ count,same_srv_rate,diff_srv_rate, dst_host_same_srv_rate, dst_host_diff_srv_rate
Application Layer	Protocol_type, flag, count, srv_count, dst_host_count, dst_host_same_src_port_rate
Network Layer	Protocol_type, src_bytes, count, srv_count, dst_host_count dst_host_same_srv_rate
Transport Layer	Service, count, srv_error_rate,same_srv_rate, diff_srv_rate, dst_host_same_src_port_rate

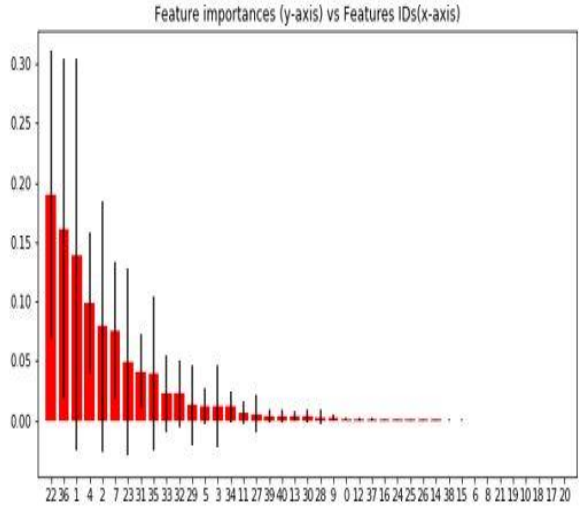


Fig.3: Feature Importance graph for all layers IDS

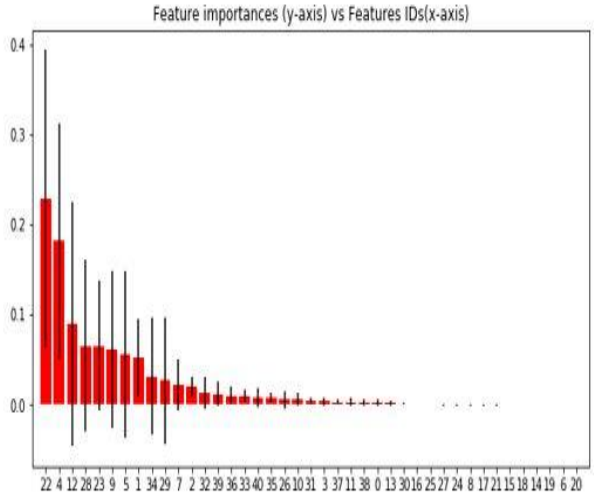


Fig.4: Feature Importance graph for Application Layer IDS

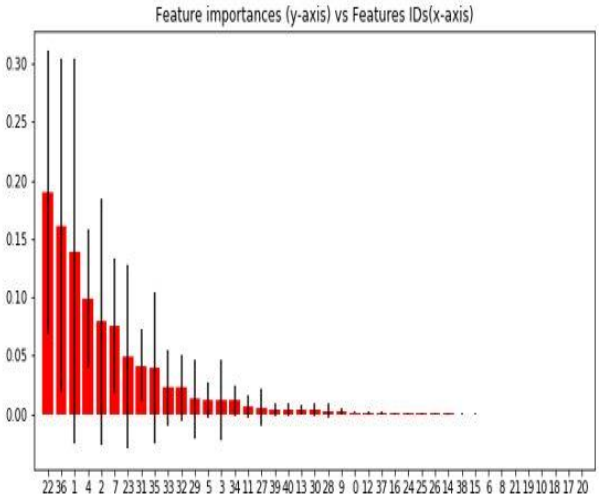


Fig.5: Feature Importance graph for Network Layer IDS

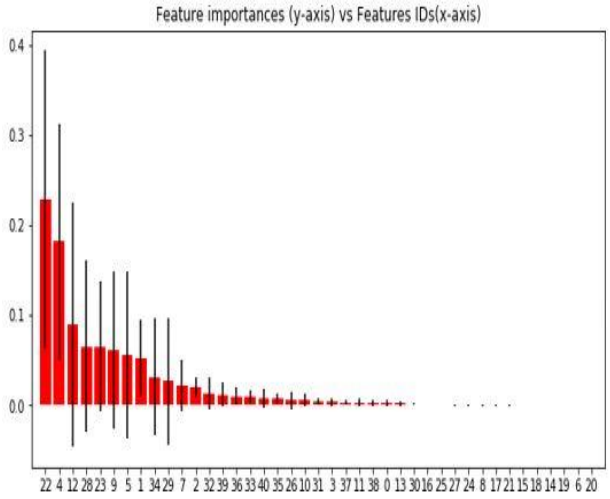


Fig.6: Feature Importance graph for Transport Layer IDS.

5.2 Evaluation metrics.

Slight change in the parameters due to some architecture requirements is the way of finding out how effective is the model based on metric and datasets. Different performance metrics are used to evaluate the performance of the IDS classifiers by tuning the hyper-parameters of the BRNN algorithm. The same method and type of experiments were conducted on each IDS classifier (All layers, application layer, transport layer, and network layer classifiers). To get more insight into the model. The value of training accuracy, recall and false alarm rate with learning rate and time-steps were then compared to understand the behavior of model with change in hyper-parameters.

5.3 Performance results of all-layer ids classifier:

The experiment was done using different input sets of hyper-parameters with the purpose of finding out which hyper-parameters will have the best impact on the model in terms of accuracy, recall and F1 Score. After selecting the time-step we searched for the learning-rate which produces best training accuracy. The detailed results are shown in Table 3.

Based on the results from the above table, it can be inferred that the model performance is optimized when the input is given with '10' time-steps and thus, this value is selected for further experiments for the All-Layers IDS in the research.

Table 3: Evaluation Metrics for All Layer IDS Classifier.

Time steps	Train Accuracy	Precision	Recall	F1 Score
10	99.949	1	0.999	99.974
20	98.592	1	0.985	99.291
30	94.616	1	0.946	97.233
40	99.744	1	0.997	99.872
50	99.432	1	0.994	99.528
60	99.6	1	0.996	0.997
70	98.123	1	0.981	0.9876
80	99.72	1	0.997	0.9974
90	98.821	1	0.988	0.99189
100	99.65	1	0.999	0.998

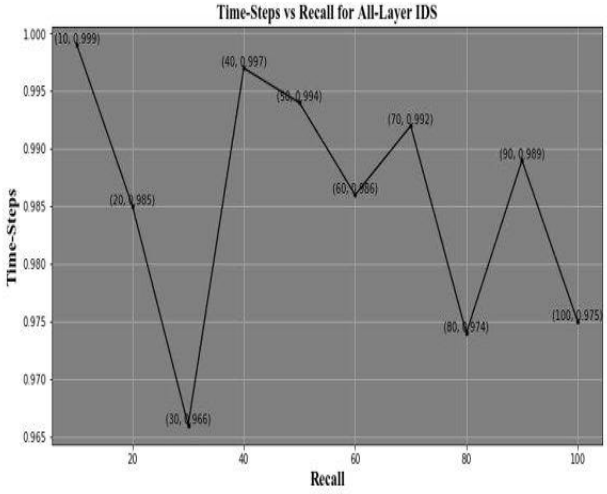


Fig.7: Impact of time-steps on recall in All-Layer IDS classifier.

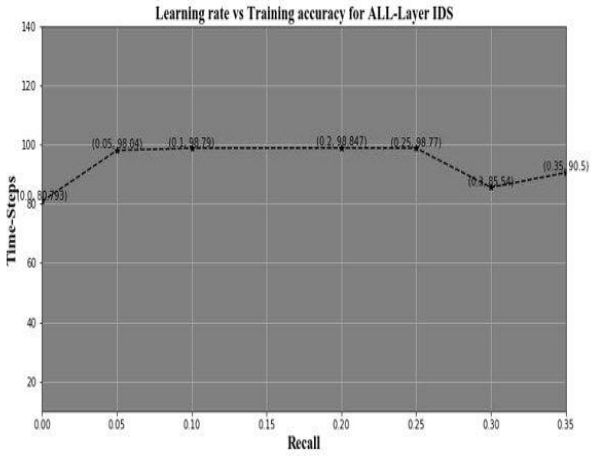


Fig.8: Impact of learning rate over training accuracy in All-Layer IDS classifier.

5.3.1 Performance results of application layer ids classifier.

This experiment was conducted on dataset with attacks that occurs on application layer and the best accuracy recorded at '60' time steps. Based on the results from the above table it is seen that the model performance is optimized when the input given is 60 time- steps . the results can be interpreted in Table 4 and Figures 9 and 10.

Table 4: Evaluation Metrics for Application Layer IDS classifier.

Time steps	Train Accuracy	Precision	Recall	F1 Score
10	96.832	1	0.968	0.983
20	93.077	1	0.930	0.964
30	98.374	1	0.983	0.991
40	96.426	1	0.964	0.981
50	98.306	1	0.983	0.991
60	99.587	1	0.995	0.997
70	87.973	1	0.897	0.936
80	94.215	1	0.942	0.970
90	86.118	1	0.861	0.925
100	97.156	1	0.971	0.985

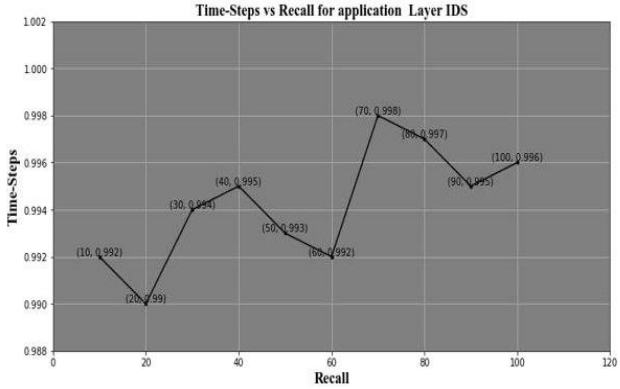


Fig.9: Impact of time-steps on recall in Application-Layer IDS classifier.

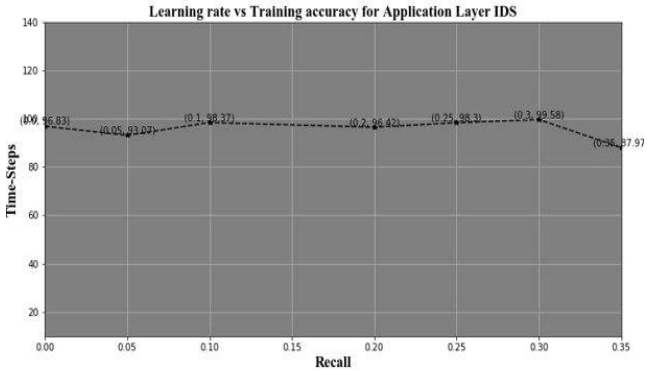


Fig.10: Impact of learning rate over training accuracy in the Application-Layer IDS classifier.

5.3.2 Performance results of transport-layer ids classifier
With a slight change in regards to learning rate, the best training accuracy was recorded given the input of 20-time step and 0.001 learning rate. All the experiments are performed on the transport layer data set which contains those

Table 5: Evaluation Metrics for Transport Layer IDS classifier.

Time steps	Train Accuracy	Precision	Recall	F1 Score
10	89.743	1	0.992	0.945
20	99.949	1	0.989	0.999
30	99.847	1	0.994	0.999
40	93.619	1	0.995	0.967
50	95.536	1	0.993	0.977
60	99.476	1	0.992	0.996
70	99.45	1	0.998	0.995
80	98.83	1	0.982	0.99
90	98.65	1	0.995	0.988
100	99.234	1	0.996	0.99

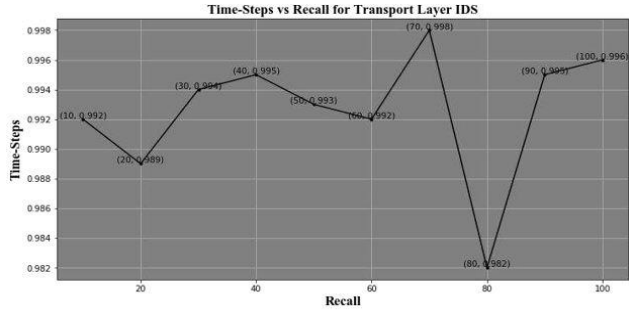


Fig.11: Impact of time-steps on recall in Transport-Layer IDS classifier

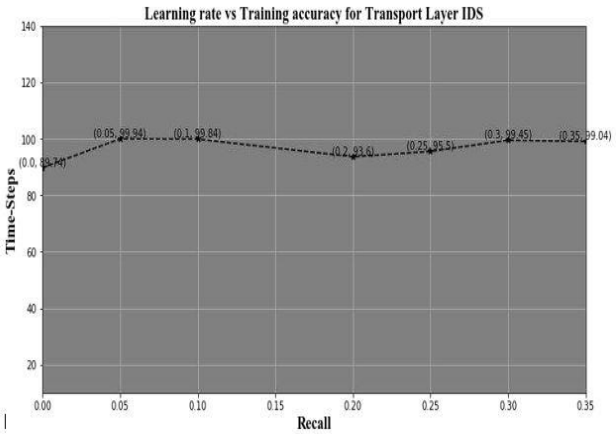


***Fig.12*: Impact of learning rate over training accuracy in Transport-Layer IDS classifier.**

Based on the results from the above table, it can be inferred that the model performance is optimized when the input is given with '20' time-steps.

5.3.3 Performance results of network-layer ids classifier:
In this experiment, the dataset with intrusion attacks that occurs on network layer was used to conduct experiments and the best training accuracy was recorded given the input of 20 times steps and 0.001 for learning rate. Based on the results from the above table, it can be inferred that the model performance is optimized when the input is given with '20' time-steps and 0.001 learning rate.

Table 6: Evaluation Metrics for Network Layer IDS classifier.

Time steps	Train Accuracy	Precision	Recall	F1 Score
10	98.793	1	0.987	0.993
20	99.542	1	0.995	0.997
30	99.035	1	0.993	0.99515
40	99.132	1	0.991	0.9956
50	99.025	1	0.990	0.99510
60	99.45	1	0.992	0.996
70	99.44	1	0.999	0.999
80	99.41	1	0.992	0.996
90	99.46	1	0.992	0.996
100	99.41	1	0.995	0.995

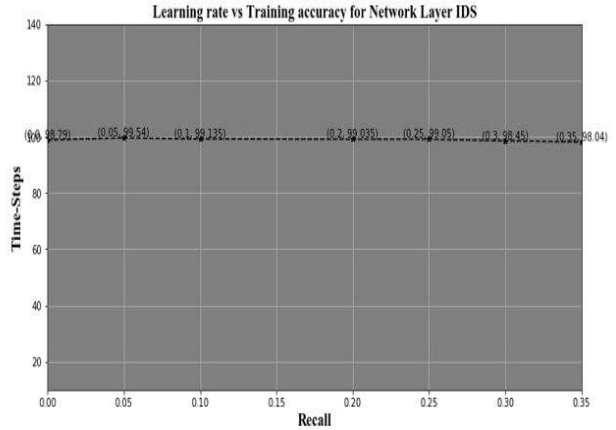


Fig.13: Impact of time-steps on recall in Network-Layer IDS classifier

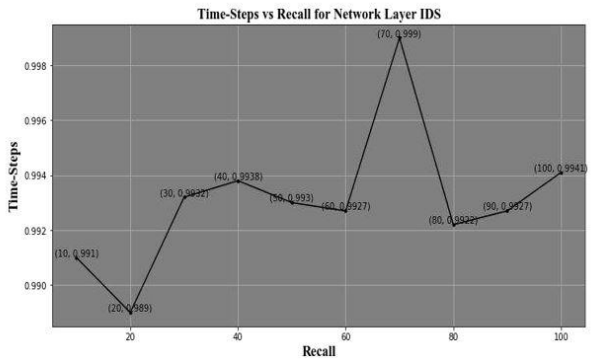


Fig.14: Impact of learning rate over training accuracy in Network-Layer IDS classifier.

5.4 Comparison of the results for IDS classifiers:

For all the IDS classifiers the optimal performance was obtained by tuning hyperparameters with the learning rate of 0.001 but with different time steps since we are dealing with imbalanced classification problem where different metrics comes into play.

Accuracy is the best way to assess the model but yet it is necessary to maximize the recall because it helps the model to find all the relevant cases within a dataset. A balanced classification model with the optimal balance of recall and precision was created where F1 score comes into play.

5.5 Comparison of the IDS classifiers performance with existing work

To assess the novelty and contribution of the current research, current results were compared with existing work performed by machine learning algorithms on intrusion detection classification as seen in Table 7. It can be observed that the current research has out-performed the performances of all the existing work.

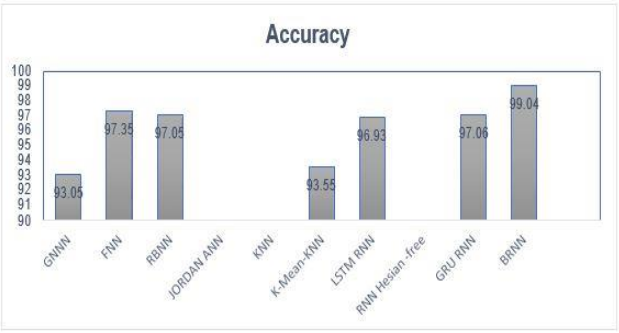


Fig.15: Comparisons of existing IDS classifiers to the proposed IDS classifiers.

VI. CONCLUSION AND FUTURE WORK.

The security necessity of the IoT is deemed to become even more important in the future based on how IoT is evolving. The progress of increasing connectivity of IoT devices combined with long connectivity without human intervention and unprecedently evolving network threats, smart security solutions which can cope with that are needed.

The results of this research revealed that BRNN stands up and outperforms other algorithms like normal RNN and GRNN with the accuracy of 99.04% .It overcomes the gap missing from Gated recurrent neural network and RNN by adding more cells and hidden neurons to allow get information from past and future states in contrary to the RNN model that has a major drawback called the vanishing gradient and GRU because it has less cell compared to the BRNN.

For future work, one can evaluate further architectures that deal with multimodal data on the intrusion detection dataset for IoT. Moreover, the aim is to investigate the application of different architectures in one framework, as well as deploying these techniques in IoT applications to develop robust security solutions using the full KDD Cup'99 dataset. This research can be taken further to devices with huge processing power and huge amount of real time data since the IoT is revolutionizing every single aspects of our life security issues also should be addressed in order to maximize IoT advantages and this is possible with artificial intelligence.

REFERENCES

[1] (2020). Retrieved 1 April 2020, from http://kdd.ics.uci.edu/databases/kddcup99 opened april13th.2019.

[2] Ammar, A. (2015). A Decision Tree Classifier for Intrusion Detection Priority Tagging. Journal of Computer And Communications, 03(04), 52-58. https://doi.org/10.4236/jcc.2015.34006

[3] Cs.columbia.edu. (2020). Retrieved 1 April 2020, from http://www.cs.columbia.edu/~mcollins/ibm12.pdf.

[4] Decision-Making, M., Tech, T., FinTech, T., A Women in Tech Study: Motivation, R., Healthcare, T., & Tech, I. et al. (2020). Techopedia - Where IT and Business Meet. Techopedia.com. Retrieved 1 April 2020, from https://www.techopedia.com.

[5] Handa, A., Sharma, A., & Shukla, S. (2019). Machine learning in cybersecurity: A review. Wires Data Mining And Knowledge Discovery, 9(4). https://doi.org/10.1002/widm.1306

[6] Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A., & Jaitly, N. et al. (2012). Deep Neural Network's for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. IEEE Signal Processing Magazine, 29(6), 82-97. https://doi.org/10.1109/msp.2012.2205597

[7] Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

[8] IEEE International Conference on Acoustics, Speech, and Signal Processing. (2008), 56(8), 4110-4110. https://doi.org/10.1109/tsp.2008.928592

[9] Internet of Things (IoT) news, blogs and analysis - IoTAgenda.com. Internetofthingsagenda.techtarget.com. (2020). Retrieved 1 April 2020, from https://internetofthingsagenda.techtarget.com.

[10] Investopedia. Investopedia. (2020). Retrieved 1 April 2020, from https://www.investopedia.com.

[11] Jia, Y., Wang, M., & Wang, Y. (2019). Network intrusion detection algorithm based on deep neural network's. IET Information Security, 13(1), 48-53. https://doi.org/10.1049/iet-ifs.2018.5258

[12] Jiang, F., Fu, Y., Gupta, B., Lou, F., Rho, S., Meng, F., & Tian, Z. (2018). Deep Learning based Multi-channel intelligent attack detection for Data Security. IEEE Transactions on Sustainable Computing, 1-1. https://doi.org/10.1109/tsusc.2018.2793284

[13] Samsung Next. Samsung NEXT. (2020). Retrieved 1 April 2020, from https://samsungnext.com.
[14] U.Farooq, M., Waseem, M., Khairi, A., & Mazhar, S. (2015). A Critical Analysis on the Security Concerns of Internet of Things (IoT). International Journal Of Computer Applications, 111(7), 1-6. https://doi.org/10.5120/19547-1280
[15] (2020). Retrieved 4 April 2020, from https://spiritlifestyle.com/news/
[16] Ashfaq, R., Wang, X., Huang, J., Abbas, H., & He, Y. (2017). Fuzziness based semi-supervised learning approach for intrusion detection system. Information Sciences, 378, 484- 497. doi: 10.1016/j.ins.2016.04.019
[17] Breiman, L. (2001). Journal search results - Cite This For Me. Machine Learning, 45(1), 5-32. doi: 10.1023/a:1010933404324
Gori, M., Maggini, M., & Rossi, A. (2016). Neural network training as a dissipative process. Neural Network's, 81, 72-80. doi: 10.1016/j.neunet.2016.05.005
[18] Gwynne, P. (2015). arXiv analysis lifts the lid on text reuse. Physics World, 28(2), 9-9. doi: 10.1088/2058- 7058/28/2/17
[19] Hasan, M., Nasser, M., Ahmad, S., & Molla, K. (2016). Feature Selection for Intrusion Detection Using Random Forest. Journal Of Information Security, 07(03), 129-140. doi: 10.4236/jis.2016.73009
[20] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444. doi: 10.1038/nature14539
[21] Li, Y., & Guo, L. (2007). An active learning based TCM-KNN algorithm for supervised network intrusion detection. Computers & Security, 26(7-8), 459-467. doi: 10.1016/j.cose.2007.10.002
[22] Mukkamala, S., Sung, A., & Abraham, A. (2005). Intrusion detection using an ensemble of intelligent paradigms. Journal Of Network And Computer Applications, 28(2), 167-182. doi: 10.1016/j.jnca.2004.01.003
[23] Obeidat, I., Hamadneh, N., Alkasassbeh, M., Almseidin, M., & AlZubi, M. (2019). Intensive Pre-Processing of KDD Cup
99 for Network Intrusion Classification Using Machine Learning Techniques. International Journal Of Interactive Mobile Technologies (Ijim), 13(01), 70. doi: 10.3991/ijim.v13i01.9679
[24] Protić, D. (2018). Review of KDD Cup '99, NSL-KDD and Kyoto 2006+ datasets. Vojnotehnicki Glasnik, 66(3), 580-596. doi: 10.5937/vojtehg66-16670
[25] Rai, N., & Chansarkar, S. (2017). Cyberspace Security : An Overview for Beginners. Defence Science Journal, 67(4), 483. doi: 10.14429/dsj.67.11542
[26] S, D., & S, R. (2014). PERFORMANCE COMPARISON FOR INTRUSION DETECTION SYSTEM USING NEURAL NETWORK WITH KDD
[27] LeCun, Y., Bengio, Y. and Hinton, G., 2015. Deep learning. Nature, 521(7553), pp.436-444.
[28] Murata, N., Yoshizawa, S. and Amari, S., 1994. Network information criterion-determining the number of hidden units for an artificial neural network model. IEEE Transactions on Neural Network's, 5(6), pp.865-872.
[29] Xu, K., Wang, X., Wei, W., Song, H. and Mao, B., 2016. Toward software defined smart home. IEEE Communications Magazine, 54(5), pp.116-122.
[30] Hasan, M., Nasser, M., Ahmad, S. and Molla, K., 2016. Feature Selection for Intrusion Detection Using Random Forest. Journal of Information Security, 07(03), pp.129-140.

