A novel approach to ensemble MLP and random forest for network security

Bhushan Deore[1] , *Aditya* Kyatham[1] *and Shubham* Narkhede[1]

[1]Electronics and Telecommunication, Ramrao Adik Institute of Technology, Nerul-400706, Navi Mumbai, India

Abstract. The following paper provides a novel approach for Network Intrusion Detection System using Machine Learning and Deep Learning. This approach uses two MLP (Multi-Layer Perceptron) models one having 3 layers and other having 6 layers. Random Forest is also used for classification. These models are ensembled in such a way that the final accuracy is boosted and also the testing time is reduced. Researchers have implemented various ways for the ensemble of multiple models but we are using contradiction management concept to ensemble machine learning models. Contradiction Management concept means if two machine learning models are contradicting in their decisions (in our case 3-layer MLP and Random Forest), then the third model's (6-layer MLP) decision is considered whose accuracy is higher than the previous models. The third model is only used for testing when the previous two models contradict in their decision because the testing time of third model is higher than the two previous models as the third model has complex architecture. This approach increased the final accuracy as ensemble of multiple models is done and also testing time has reduced. The novelty of this paper is the choice and the combination of the models for the purpose of Network security.

1. INTRODUCTION

Today, use of Internet has drastically increased. In future, every person on the earth would be a part of network. Many internet users are not aware about the hacking techniques. So, there is high probability of intrusion of a black hat hacker into a personal network of various devices. Many companies are using cloud technology for their system operations. If an unauthorized intruder gets access to the private data of the company, then the intruder could use the data for gaining money.

Network Intrusion Detection system (NIDS) is a very important part of a network. Along with the firewall, NIDS has to be deployed because firewall blocks a particular website only if the website is blacklisted in the database. But NIDS works on the packet data. It traces the packet and uses anomaly detector to detect the intruder.

In this approach, we used Machine Learning models to detect the intruder. The decisions of multiple machine learning models are considered and using the proposed approach decision is finalized. The packet will be captured by the packet capturer and then the packet is tested on the trained Machine Learning models. The model gives its decision. If the decision is positive i.e. the packet is anomaly, then the system administrator is alerted about the packet. This approach also decreases the packet waiting time. Packet waiting time means the time for which the packet is tested by the NIDS.

The main goal of the paper is to boost the accuracy of NIDS using multiple machine learning models and also to reduce the packet waiting time. We have used CIC (Canadian Institute of Cybersecurity) dataset for training the model. The packet analyzer used is CIC-Flow-Meter.

2. RELATED WORK

YONG ZHANG et al [1] introduced a new network intrusion detection model named the deep hierarchical network, that combines LeNet-5 and LSTM Neural networks.

Meng Wang et al [2] proposed a method to choose optimal feature for detecting DDoS attack using sequential feature selection with MLP.

Yi Yi Aung et al [3] explained the method of applying Random Forest for Network Intrusion Detection System.

Prachi Barapatre et al [4] used MLP with back-propagation algorithm to classify attacks. Dataset used was KDD99 dataset. They analysed the working of MLP model and studied the advantages and disadvantages of MLP model. The main goal was to decrease false alarms.

Dimitra Chamou et al [5] used Deep Neural Networks for classification of DDoS attacks and Malware attacks. They used 5-layer which are fully-connected layers of neural network.

Vinayakumar et al [6] explained the working of Convolutional Neural Networks (CNN) in Network Intrusion Detection System. They applied the data on various models having various layers.

3. PROPOSED APPROACH

3.1 Dataset Introduction: -

We trained our algorithm on the CIC (Canadian Institute for Cyber Security) datasets of 5 days. The network data has been captured using the CICFlowMeter which is a packet capturer developed by CIC. This dataset has updated attacks compared to NSL-KDD dataset. The data was captured in 2017 by CIC.

This dataset is built on the abstract behaviour of 25 users based on HTTP, HTTPS, FTP, SSH and email protocols. The capturing of the network data started from 9 am Monday July 3rd 2017, and ended on Friday 7th July,2017 at 5 pm. Monday was a normal traffic so we didn't use it for training. We used the dataset of the other 4 days. On Tuesday = SSH-Patator and FTP-Patator, Wednesday=DoS/DDOS, Thursday=Infiltration, Friday =Portscan, DDOS, Bot attacks were done. For more information about dataset and packet capturer refer [13].

All the datasets contain 79 features including label column and samples are over 50000, non-null and data types consist of int, float and object. The final dataset consists of packets from all the 4 days with labels as Anomaly and Normal which basically can be used for Anomaly Detector.

3.2 Feature Engineering: -

In feature engineering, we need to select the important features which are responsible for classification. Because some features are not relevant to the output label which decreases accuracy. Those irrelevant features should be dropped out from the dataset. To calculate the relevance of the features with the output label, we should calculate correlation of each and every feature with the output label. If the correlation of a particular feature is less, then those particular features should be dropped off from the dataset. Feature importance in the features of CIC Dataset (top 25 are shown but top 58 were considered for training) shown in Fig. 1.

After feature reduction or dimensionality reduction, the data should be split into training, testing and validation sets. The splitting ratio was 80% training, 10% testing and 10% validation. Now, the data should be label encoded. Label encoding is a process in which the character type features or features containing character type data, should be converted to categorical data

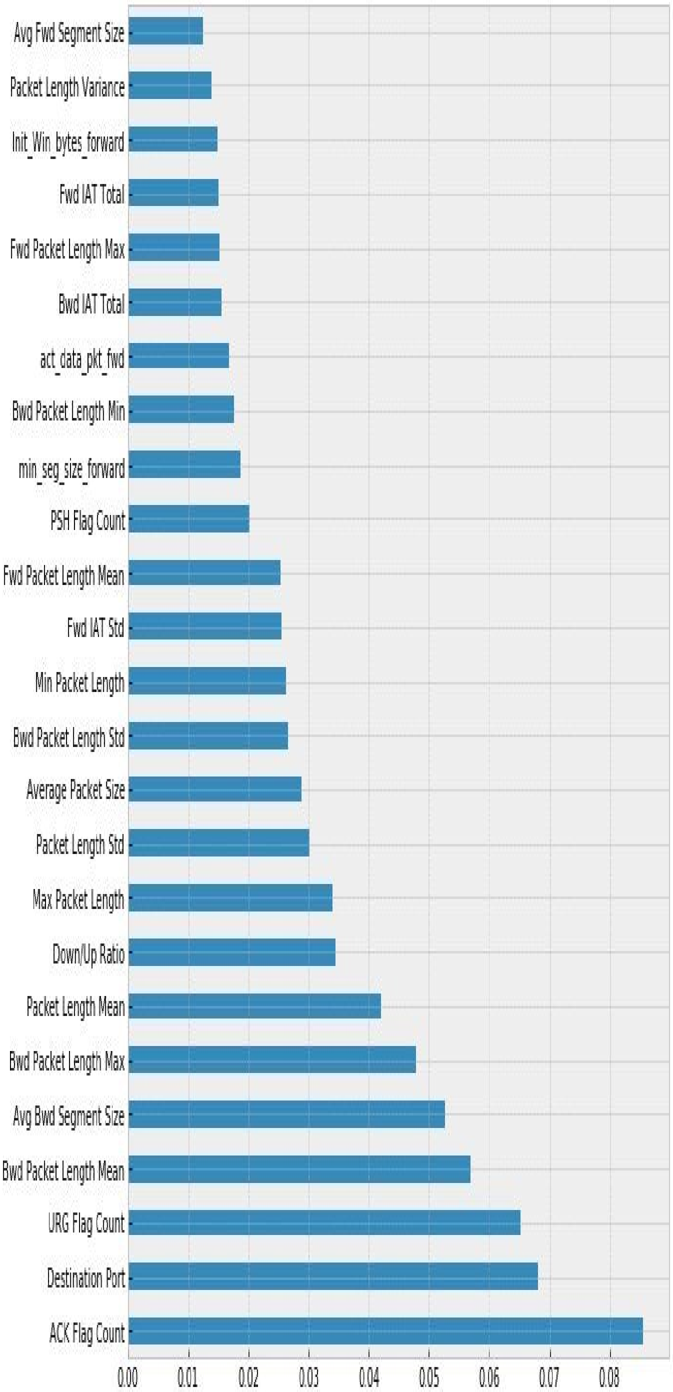

Fig. 1. Feature Importances

Now, the data has to be normalized to prevent biasing condition and then it has to be reshaped to a particular array which can be fed to Neural Networks.

3.3 Model Building: -

The CIC dataset is used to train three models. Three models are: - MLP (Multi-Layer Perceptron) with 3-layer, 6-layer MLP and Random Forest. After training, the packet will be sent to Random Forest and 3-layer MLP for testing. Both the models will give their decisions. Now, both the decisions are compared and if there is contradiction then that particular packet is sent to 6-layer MLP. This 6-layer MLP has more accuracy than Random Forest and 3-layer MLP but more testing time than the both. So, as the 6-layer MLP has more testing time, testing all the packets on 6-layer MLP would increase packet waiting time which is not desirable. There is no need to test a packet on 6-layer MLP if the 3-layer MLP and Random Forest both are giving same decision because this shows that the probability of decision being correct is high as both the models gave same decision.

If all the packets are tested on 6-layer MLP then the testing time will be high but, in our approach, we found out that only few samples (<5 %) are sent to 6-layer MLP which decreases the overall testing time. The 3-layer MLP has 128 neurons in 1st and 3rd layer and then 64 neurons in 2nd layer with 'relu' activation and output activation as 'sigmoid'. The dropout is 1%. Similar, configuration is there for 6-layer MLP with alternate layers of 128 and 64 neurons. Architecture is shown in Fig. 4.

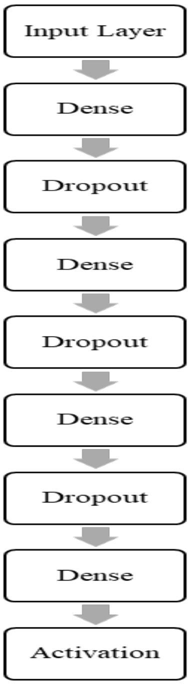

Fig. 2 3-Layer Multi-Layer Perceptron (MLP) Model

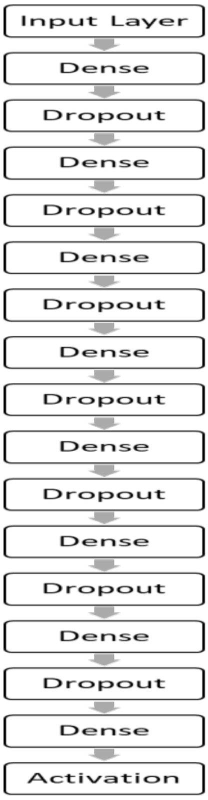

Fig. 3 6-Layer MLP Model

For Random Forest, we used 50 estimators and the criterion used was entropy. Rest of the parameters were set to default.

Below is the architecture of the proposed approach:

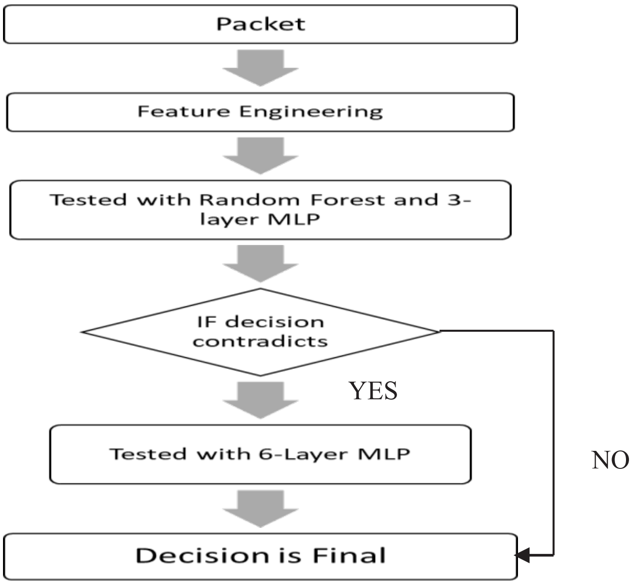

Fig. 4 Proposed Model Architecture of an Anomaly Detector.

4. RESULT ANALYSIS

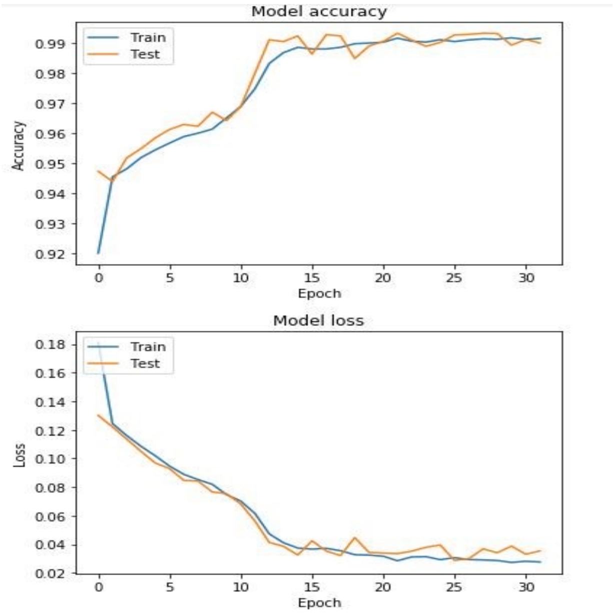

Fig. 5 Loss and Accuracy graph of 3-layer MLP

Fig. 5 & 6 shows the graph of loss decreasing and accuracy increasing which states that the model is successfully learning. There are some ripples during the learning process but ultimately the ripples decrease signifying that the model is learning with stable state.

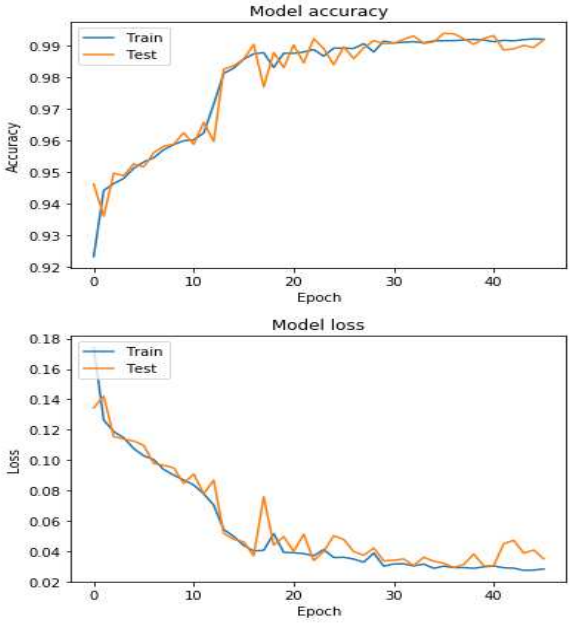

Fig. 6 6-Layer MLP Loss & Accuracy Graph

TABLE I. 3-layer MLP Metrics

Labels	Accuracy	Precision	Recall	F1 score
ANOMALY	98.87	97.00	99.00	98.00
NORMAL		100.00	99.00	99.00

TABLE. II. Random Forest Metrics

Labels	Accuracy	Precision	Recall	F1 score
ANOMALY	85.32	100.00	51.00	68.00
NORMAL		83.00	100.00	91.00

TABLE III. 6-Layer MLP Metrics

Labels	Accuracy	Precision	Recall	F1 score
ANOMALY	99.12	99.00	98.00	99.00
NORMAL		99.00	99.00	99.00

TABLE IV. Proposed Approach Metrics

Labels	Accuracy	Precision	Recall	F1 score
ANOMALY	99.31	99.00	99.00	99.00
NORMAL		100.00	100.00	100.00

From the above metrics, we can see that proposed approach metrics has more accuracy than the other models as it boosted the accuracy by ensemble of the above existing models. Table I is the result, when the dataset is tested only on 3-Layer MLP, which tells us that it gives a pretty good metrics. As the dataset is little imbalanced with respect to the labels, we also considered the precision, recall and F1-score. Table II is for Random Forest in which metrics are lower than 3-layer MLP but can be used as a validation model for 3-layer MLP. Table III is for 6-layer MLP, as layers are more the learning is more accurate, so in this case the metrics are higher. Now, using the models in the table I, II and III in the proposed manner, the results show that the metrics are boosted. The testing time is also found less when the packet is tested with Random Forest and 3- Layer MLP as compared to 6-Layer MLP as more the layers, more will be the mathematical calculations and thus more testing time is needed. The drawback would be when the packet has to be tested for all the 3 models, but the probability of that happening is found to be very less i.e. 5%.

Following are the testing times:

Models	Testing Time (seconds)
Random Forest + 3-layer MLP	0.256
6- Layer MLP	0.264

Comparison of different algorithms

Sr. No	**Algorithm**	**Accuracy**	**Precision**	**Recall**	**F1 Score**
1	Nathan Shone's algorithm	91.13	89.00	91.00	90.00
2	Auto Encoder with Neural Network	90.15	91.85	90.64	88.97
3	Adaboost with Neural Network	94.08	95.05	94.08	94.42
4	Bagging Neural Network	95.56	95.81	95.56	95.48
5	Random Forest	94.58	94.00	95.00	94.00
6	LDA (Linear Discriminant Analysis)	96.05	96.00	96.00	96.00
7	QDA (Quadratic Discriminant Analysis)	93.10	92.00	93.00	93.00
8	Linear SVC	94.08	94.00	94.00	94.00
9	Simple RNN (Recurrent Neural Network)	95.07	96.42	95.82	96.11
10	RNN-LSTM	97.04	96.41	94.39	95.38
11	GRU	94.09	97.00	96.00	97.00
12	Simple CNN	96.07	97.00	96.00	97.00
13	KNN	94.11	93.00	94.00	94.00
14	**The proposed Model**	99.31	99.50	99.50	99.50

We used ensembled algorithm i.e. Random Forest and Neural Network that's why our metrics is higher than others.

5. CONCLUSION

We can conclude that, by considering decisions of multiple models and if the decisions contradict, then that decision will be taken by another Neural Network model which has complex architecture but has high accuracy. In this way, the resulting model will have high accuracy and low testing time. In the future work, there is scope of developing new techniques for ensemble modelling

References

[1] Zhang, Yong, et al. "Network intrusion detection: Based on deep hierarchical network and original flow data." *IEEE Access* 7 (2019): 37004-37016.

[2] Wang, Meng, Yiqin Lu, and Jiancheng Qin. "A dynamic MLP-based DDoS attack detection method using feature selection and feedback." *Computers & Security* 88 (2020): 101645.

[3] Yi, Y. A., and Myat Myat Min. "An analysis of random forest algorithm based network intrusion detection system." *Proceedings of the International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, Kanazawa, Japan*. 2017.

[4] Barapatre, Prachi, et al. "Training MLP neural network to reduce false alerts in IDS." *2008 International Conference on Computing, Communication and Networking*. IEEE, 2008.

[5] Chamou, Dimitra, et al. "Intrusion Detection System Based on Network Traffic Using Deep Neural Networks." *2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*. IEEE, 2019.

[6] Vinayakumar, R., K. P. Soman, and Prabaharan Poornachandran. "Applying convolutional neural network for network intrusion detection." *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2017.

[7] Ertam, Fatih, and Orhan Yaman. "Intrusion detection in computer networks via machine learning algorithms." *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*. IEEE, 2017.

[8] Esmaily, Jamal, Reza Moradinezhad, and Jamal Ghasemi. "Intrusion detection system based on multi-layer perceptron neural networks and decision tree." *2015 7th Conference on Information and Knowledge Technology (IKT)*. IEEE, 2015.

[9] Ahmad, Iftikhar, et al. "Intrusion detection using feature subset selection based on MLP." *Sci Res Essays* 6.34 (2011): 6804-6810.

[10] Yulianto, Arif, Parman Sukarno, and Novian Anggis Suwastika. "Improving adaboost-based intrusion detection system (IDS) performance on CIC IDS 2017 dataset." *Journal of Physics: Conference Series*. Vol. 1192. No. 1. IOP Publishing, 2019.

[11] Sathesh, A. "ENHANCED SOFT COMPUTING APPROACHES FOR INTRUSION DETECTION SCHEMES IN SOCIAL MEDIA NETWORKS." *Journal of Soft Computing Paradigm (JSCP)* 1.02 (2019): 69-79.

[12] Anguraj, Dinesh Kumar, and S. Smys. "Trust-based intrusion detection and clustering approach for wireless body area networks." *Wireless Personal Communications* 104.1 (2019): 1-20.

[13] Dhanabal, L., and S. P. Shantharajah. "A study on NSL-KDD dataset for intrusion detection system based on classification algorithms." *International Journal of Advanced Research in Computer and Communication Engineering* 4.6 (2015): 446-452.

[14] Shone, Nathan, et al. "A deep learning approach to network intrusion detection." *IEEE Transactions on Emerging Topics in Computational Intelligence* 2.1 (2018): 41-50.

[15] CIC dataset :https://www.unb.ca/cic/datasets/ids-2017.htm