Assessment of Machine Learning Algorithms for Network Intrusion Detection

Mayur Sonthalia, Jayavignesh Thyagarajan

***Abstract:** A Network Intrusion Detection System (NIDS) is a framework to identify network interruptions as well as abuse by checking network traffic movement and classifying it as either typical or strange. Numerous Intrusion Detection Systems have been implemented using simulated datasets like KDD'99 intrusion dataset but none of them uses a real time dataset. The proposed work performs and assesses tests to overview distinctive machine learning models reliant on KDD'99 intrusion dataset and an ongoing created dataset. The machine learning models achieved to compute required performance metrics so as to assess the chosen classifiers. The emphasis was on the accuracy metric so as to improve the recognition pace of the interruption identification framework. The actualized calculations showed that the decision tree classifier accomplished the most noteworthy estimation of accuracy while the logistic regression classifier has accomplished the least estimation of exactness for both of the datasets utilized.*

***Keywords**: Intrusion Detection System, Machine Learning, Network Attacks, Real time*

I. INTRODUCTION

With the brisk headway of data innovation in the past two decades, various computer frameworks are comprehensively used by industries, businesses and various fields of the human life. Subsequently, building strong frameworks is a huge task for IT regulators. Then again, the fast advancement of data innovation delivered a few difficulties to construct solid systems which is an exceptionally troublesome errand. There are numerous kinds of attacks undermining the confidentiality of computer systems. No firewall is secure, and no framework is immune. Aggressors interminably develop new undertakings and attack frameworks proposed to sidestep shields. Various attacks impact other malware or social structure to get customer affirmations that grant them access to the framework and data. A Network Intrusion Detection System (NIDS) is basic for security since it engages a framework to distinguish and respond to toxic traffic.

The principle job of a network intrusion detection system is to promise IT work power is prompted when an ambush or framework interference might be happening. A Network Intrusion Detection System (NIDS) assess the traffic on the framework as data goes between structures inside the framework. The NIDS screen organize traffic and triggers alerts when suspicious development or acknowledged threats are distinguished, so IT staff can take a gander at even more eagerly and figure out how to stop the intrusion. The major network attacks which comes into the scenario are Denial of Service attack [1] in which the attacker send innumerable requests to the server and the legitimate user is denied the service, User to Root attack in which an attacker (a user) tries to explore the vulnerabilities in the system by gaining its root access, Remote to Local attack where an aggressor attempts to get the local hold in the system, sending malicious packets from a remote system and probing attack in which the attacker initially gathers information about the security controls of the system and then try to explore the vulnerabilities in it. The work proposed in this paper makes use of a real time dataset which is generated in due course of the work to estimate how the machine learning models will perform in the real time scenario. The objective of the work is to automatically generate the rules for network intrusion detection system using machine learning algorithms and measure its performance metrics based on a real time dataset. This is done by applying the algorithms on the dataset. The processing of this dataset is done based on the features and accordingly the rules for intrusion detection is generated. The proposed work finds the best suitable machine learning algorithm for an efficient intrusion detection system.

II. LITERATURE SURVEY

As per the study carried out in [2], the authors made use of the KDD'99 intrusion dataset and actualized the pre-process stages for example standardization of the properties and changing over representative characteristics. Neural network feed forward was actualized in multiple papers. The authors have commented that the feed forward neural system isn't reasonable enough for Root to Local (R2L) and User to Root (U2R) assaults however then again, it has been proved worthy for DoS as well as PROBE assaults. As it identifies to execute neural network against KDD'99 interruptions, the exertion of [3] the authors prevailing to actualize the accompanying four calculations: Fuzzy ARTMAP, Radial-based Function, Back proliferation (BP) and Perceptron-back spread crossover (PBH). The four calculations assessed and tried for interruptions discovery the BP and PBH calculations recorded most noteworthy accuracy rate. In [4] the authors select the features that are of utmost importance using the KDD

Revised Manuscript Received on April 18, 2020.

* Correspondence Author

Mayur Sonthalia, Electronics and Computer Engineering, School of Electronics Engineering, Vellore Institute of Technology, Chennai, India. Email: msonthalia672@gmail.com

Jayavignesh Thyagarajan*, School of Electronics Engineering, Vellore Institute of Technology, Chennai, India. Email: jayavignesh.t@vit.ac.in

'99 intrusion dataset which is a simulated dataset to reduce computation time and increase accuracy. The focus of the work is to reduce false positive rate by using neural network. On the other hand, in [5] the algorithm used was based on the information gain of the system. The authors used multivariate method for the detection of the DoS assault. The entirety of the past research works had a regarded commitment and simultaneously show that KDD'99 intrusion dataset gives the mentioned condition to test and assess different ML calculations. Likewise, the past works present that the single segregated ML calculation would not propose the acknowledged detection rate. Right now, following ML classifiers (Decision Tree, Naive Bayes, and Logistic Regression) were actualized, tried and assessed dependent on KDD'99 intrusion dataset, Kaggle dataset and a real-time dataset generated for the purpose.

The authors of [6] proposed genetic algorithm calculation for network intrusion detection. False positive rate acquired with Genetic Algorithm is 0.3046 which is excessively high as concluded by the authors. [7] proposes a system that supports up to 23.76 Gbps interface speed however the network intrusion detection framework ceaselessly utilizes extra assets in the framework it is observing in any event, when there are no interruptions happening. The authors of [8] proposed a work that actualizes SVM calculation against interruptions. The authors inferenced that SVM calculation needs excess preparing time and in view of that the convenience of the algorithm is constrained.

In [9] the authors used genetic algorithm with support vector Machine but it came out to be a failure since the considerable value of accuracy was not obtained. On the other hand, [10] proposes SM algorithms, SM obtained low accuracy but remarkable f-score. Zhi-song et. al. [11] proposes neural network but the authors themselves claims that the model fails to add more data during run time and need to be stopped when next set of data comes in. [12] proposes a packet sniffer that captures that packet and decodes it and if the decoded version is found to be malicious then the particular IP is blocked. The other works [13,14] explore deep learning approaches to build an efficient network intrusion detection system.

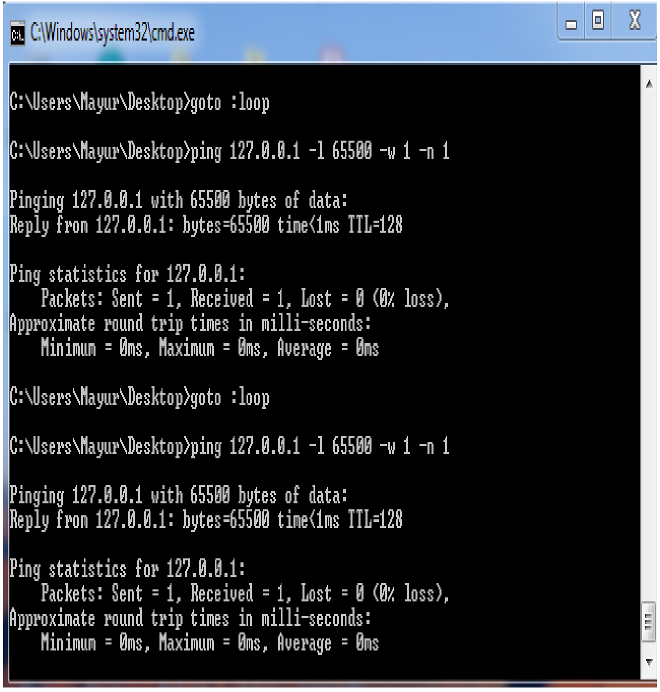
All the above work makes use of the KDD '99 intrusion dataset but the proposed work uses a real time generated dataset along with the KDD'99 intrusion dataset for the purpose of network intrusion detection.

III. IMPLEMENTATION

The machine learning algorithms are implemented to increase the efficiency of the network intrusion detection system. The proposed work makes use of three algorithms which are decision tree, logistic regression and Naïve Bayes. DoS attack is generated and the data of the packets received during DoS attack helps in the generation of a real time dataset. The collected dataset along with the KDD'99 intrusion dataset is used to measure the performance metrics of the three machine learning algorithms. The following subsections give the details of each process.

A. Generation of Ping of Death (PoD)

Ping of Death is an attack in which an attacker tries to crash or freeze the engaged PC or organization by sending twisted or inquisitively enormous bundles using a direct ping order. The size of a precisely formed IPv4 bundle including the IP header is 65,535 bytes, which incorporates 84 bytes of payload. Various true PCs basically couldn't manage greater parcels, and would crash if they receive it. This was conveniently abused in TCP/IP use in a wide extent of working structures. Since it would be the violation of internet protocol to send parcels greater than 65535 bytes, aggressors would for the most part send contorted bundles in sections. Right when the target system tries to reassemble the pieces and ends up with a bigger than normal package, the memory could get flooded and may lead to crash.

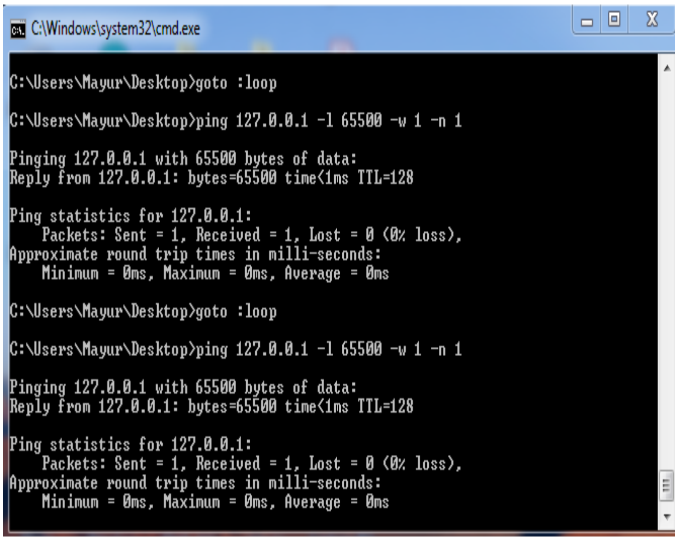


```
C:\Users\Mayur\Desktop>goto :loop

C:\Users\Mayur\Desktop>ping 127.0.0.1 -l 65500 -w 1 -n 1

Pinging 127.0.0.1 with 65500 bytes of data:
Reply from 127.0.0.1: bytes=65500 time<1ms TTL=128

Ping statistics for 127.0.0.1:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\Mayur\Desktop>goto :loop

C:\Users\Mayur\Desktop>ping 127.0.0.1 -l 65500 -w 1 -n 1

Pinging 127.0.0.1 with 65500 bytes of data:
Reply from 127.0.0.1: bytes=65500 time<1ms TTL=128

Ping statistics for 127.0.0.1:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Fig. 1.Generating Ping of Death

Ping of Death attacks were particularly fruitful considering the way that the attacker's character could be adequately personification. Additionally, a Ping of Death assailant would require no quick and dirty data on the machine he/she was attacking, except for its IP address. It is meriting note that this weakness, anyway best apparent for its abuse by PoD attacks, can truly be manhandled by whatever sends an IP datagram.

The proposed work generates Ping of Death using a C program. The program is executed to send infinite pings to the localhost and the packets are captured from the localhost using a software called Rawcap. Also, the packets are captured from the localhost when no pings are being sent. After the packets are captured, a pcap file is automatic downloaded from rawcap. This file is converted to csv file using Wireshark. The csv file is then used by machine learning algorithms for the successful computation of the required metrices.



```
C:\Users\Mayur\Desktop>goto :loop

C:\Users\Mayur\Desktop>ping 127.0.0.1 -l 65500 -w 1 -n 1

Pinging 127.0.0.1 with 65500 bytes of data:
Reply from 127.0.0.1: bytes=65500 time<1ms TTL=128

Ping statistics for 127.0.0.1:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\Mayur\Desktop>goto :loop

C:\Users\Mayur\Desktop>ping 127.0.0.1 -l 65500 -w 1 -n 1

Pinging 127.0.0.1 with 65500 bytes of data:
Reply from 127.0.0.1: bytes=65500 time<1ms TTL=128

Ping statistics for 127.0.0.1:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Fig. 2.Ping of Death data collection

B. KDD'99 Dataset

Software to recognize organize interruptions shields a computer from unapproved clients, including maybe insiders. The interruption indicator learning task is to manufacture a prescient model (for example a classifier) equipped for recognizing "awful" associations, called interruptions or assaults, and "great" ordinary associations.

The objective of 1998 DARPA program was to outline and evaluate investigate in interference distinguishing proof. A standard course of action of data to be investigated, which consolidates a wide collection of interferences re-instituted in a military framework environment, was given. The 1999 KDD interference area challenge uses a type of this dataset.

Lincoln Labs had established a situation to gain few weeks of crude TCP dump information of a neighborhood (Local Area Network) mimicking an ordinary U.S. Flying corps Local Area Network. It worked as being a genuine Air Force condition, however peppered up with different assaults.

The crude preparing information was around 4GB of packed paired TCP dump information from few weeks of system data. This was handled into around 5,000,000 association records. Likewise, the fourteen days of test information yielded around 2,000,000 association records.

The assaults are of four major types: DoS (Denial of Service), U2R (User to Root), R2L (Root to Local) and probing attack.

C. Machine Learning Algorithms

The machine learning algorithms used for the intrusion detection system are decision tree algorithm, Naïve Bayes algorithm and logistic regression algorithm. These algorithms describe how the machine learning processes the data from the dataset to classify the activities of the network as normal or anomalous. The datasets used are KDD '99 intrusion dataset and the real time generated dataset. The performance metrices include accuracy, precision, f-score and recall values. The flowchart represented in Fig. 3 represents the general workflow for the implementation of the machine learning algorithms along with the web application.

D. Web Application

A web application is developed using HTML. The HTML script has some CSS component to make the page look attractive. The web application is connected to the backend python script through a web application framework called flask.

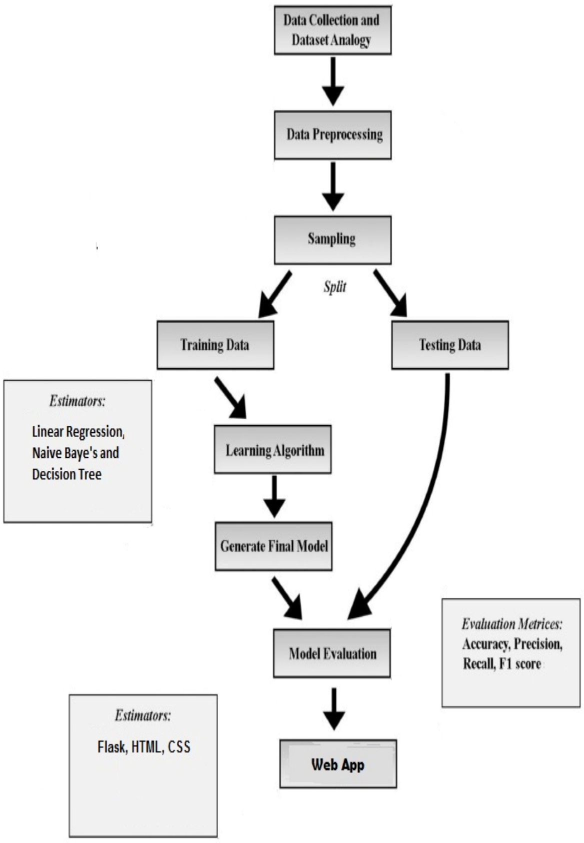
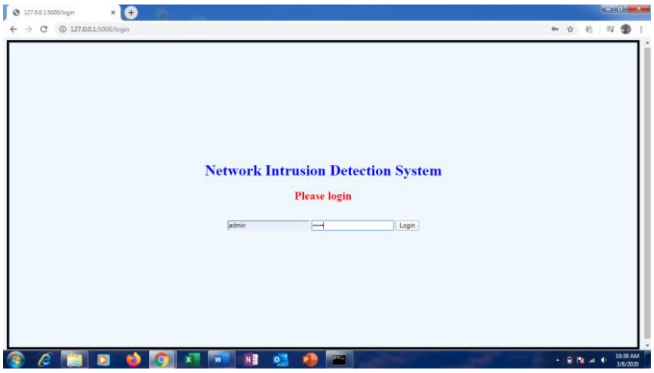


Fig. 3.Machine Learning Workflow



Network Intrusion Detection System

Please login

admin ••••• Login

Fig. 4.Web Application Login Page

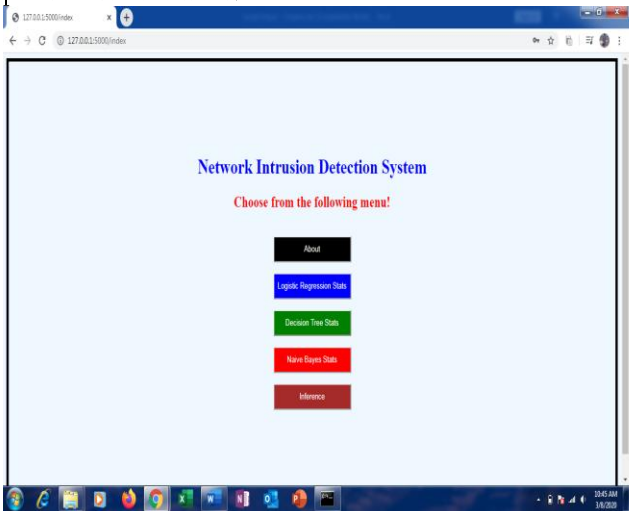
Security is implemented in the web app to prevent unauthorized access of the web app. A user can go onto the web page and sign up. After successful signup, the user can login and directly use the web application. If the user id or password entered is incorrect then the page will throw an error saying that either user id or password entered is incorrect. The login page is shown in Fig. 4.

After successful login, the user can run the machine learning programs via frontend itself and obtain the results on the frontend. The user can also learn more on the intrusion detection system and the algorithms through various information pages present in the web application. This makes it easy for the user to decide which algorithm to use for network intrusion detection.

Retrieval Number: D8689049420/2020©BEIESP
DOI: 10.35940/ijeat.D8689.049420
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Fig. 5 represents the options page through which the user can perform various tasks.



Network Intrusion Detection System

Choose from the following menu!

Fig. 5.Web Application Options Page

IV. RESULTS

The performance metrics of the machine learning models for both the datasets reveals a similar trend in the accuracies recorded. The decision tree algorithm performs the best for all the three datasets followed by Naïve Bayes and then logistic regression algorithm.

The performance metrices obtained for KDD'99 intrusion dataset is tabulated in Table 1 and the performance metrices obtained for generated dataset is tabulated in Table 2.

Table- I: KDD'99 Intrusion Dataset Results (in %)

Algorithm	Accuracy	Precision	Recall	F1-score
Decision Tree	93	99.8	91.4	95.4
Naïve Bayes	92.9	98.8	92.3	95.4
Logistic Regression	84.6	98.8	81.8	89.5

Table- II: Generated Dataset Results (in %)

Algorithm	Accuracy	Precision	Recall	F1-score
Decision Tree	72	72	72	71.9
Naïve Bayes	72	76	72.5	71.2
Logistic Regression	68	67.9	67.9	67.9

The plot of accuracy of different machine learning algorithms with each of the two datasets is shown in Fig. 6

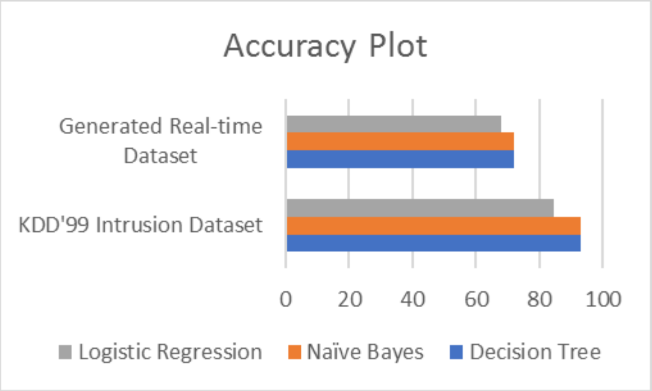


Fig. 6.Accuracy Plot

V. Conclusion

The evaluation of machine learning algorithms aims to increase the accuracy and efficiency of the Network Intrusion Detection System. In the proposed implementation and experimentation, a real time dataset was generated and the machine learning algorithms were applied on the KDD'99 intrusion dataset and the generated dataset, it is found that the decision tree algorithm performs the best as compared to Naïve Bayes and logistic regression algorithms on both the KDD'99 intrusion dataset and the real time dataset which was generated. The accuracy values observed for the KDD'99 intrusion dataset are 93%, 92.9% and 84.6% for decision tree, Naïve Bayes and logistic regression respectively. On the other hand, these values for generated dataset are 72%, 68% and 68% respectively.

REFERENCES

1. Akash & Kishan, Jai, Tiwari, Mohit & Kumar, Raj & Bharti, "Intrusion Detection System", International Journal of Technical Research and Applications, 2017.
2. S. Khanchi, F. Haddadi, V. Derhami, and M. Shetabi, "Intrusion detection and attack classification using feed-forward neural network," in Computer and Network Technology, 2010 Second International Conference on. IEEE, 2010, pp. 262–266.
3. Li, Manikopoulos, Jorgenson, Zhang, and Ucles, "Hide: a hierarchical network intrusion detection system using statistical preprocessing and neural network classification,", IEEE Workshop on Information Assurance and Security, 2001, pp. 85–90.
4. Alsharafat, "Applying artificial neural network and extended classifier system for network intrusion detection." International Arab Journal of Information Technology, 2013, vol. 10, no. 3.
5. N. Bhargava, G. Sharma, R. Bhargava, and M. Mathuria, kopoulos, "Decision tree analysis on j48 algorithm for data mining," Proceedings of International Journal of Advanced Research in Computer Science and Software Engineering, vol. 3, no. 6, 2013.
6. Venter, Pillai, Eloff, "An Approach to Implement a Network Intrusion Detection System using Genetic Algorithms", Proceedings of SAICSIT, 2004, pp:221-228.
7. D.T. Nguyen. et al., "A Reconfigurable Architecture for Network Intrusion Detection Using Principal Component Analysis", Proceedings of the International Symposium on Field Programmable Gate Arrays - FPGA'06, 2006, vol. 9, no. 4, pp. 642-645.
8. Lahre, Et. al., "Analyze different approaches for ids using kdd 99 data set," International Journal on Recent and Innovation Trends in Computing and Communication, 2013, vol. 1, no. 8, pp. 645–651.
9. Kuang, Fangjun, Weihong Xu and Siyang Zhang, "A novel hybrid KPCA and SVM with GA model for intrusion detection," Applied Soft Computing, 2014, Vol.18, pp.178-184.

Retrieval Number: D8689049420/2020©BEIESP
DOI: 10.35940/ijeat.D8689.049420
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

10. N. Shone, T. N. Ngoc, V. D. Phai and Q. Shi, "A Deep Learning Approach to Network Intrusion Detection," in IEEE Transactions on Emerging Topics in Computational Intelligence, 2018, vol. 2, no. 1, pp. 41-50.
11. Zhi-Song Pan, Song-Can Chen, Gen-Bao Hu and Dao-Qiang Zhang, "Hybrid neural network and C4.5 for misuse detection," Proceedings of the 2003 International Conference on Machine Learning and Cybernetics, 2003, Xi'an, pp. 2463-2467 Vol.4.
12. Karatas, O. Demir and O. Koray Sahingoz, "Deep Learning in Intrusion Detection Systems," 2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT), ANKARA, Turkey, 2018, pp. 113-116.
13. M. Uğurlu and İ. A. Doğru, "A Survey on Deep Learning Based Intrusion Detection System," 2019 4th International Conference on Computer Science and Engineering, Samsun, Turkey, 2019, pp. 223-228.
14. KDD cup 1999 Dataset: The UCI KDD archive, Information and Computer Science, University of California, 1999, [online], Available: http://kdd.ics.uci,edu/databases/kddcup99/kddcup99.html.

AUTHORS PROFILE

Mayur Sonthalia is a student and he is currently pursuing B.tech - Electronics and Computer Engineering, School of Electronics Engineering (SENSE), Vellore Institute of Technology (VIT), Chennai, India. His research interest includes Machine Learning, Wireless Sensor Networks (WSN) and Internet of Things (IoT).

Jayavignesh Thyagarajan is a faculty in Vellore Institute of Technology (VIT), Chennai, India. His research interests include Routing Protocols Design, Analysis of Medium Access Protocols and Congestion Control mechanism in the area of Wireless Adhoc, Sensor, Mesh, Vehicular and Opportunistic networks. He acquired his Masters in Engineering from Madras Institute of Technology, Anna University and secured First Rank with a Gold medal in Communication and Networking program. He obtained his Bachelors in Engineering degree in Electronics and Communication from College of Engineering, Guindy, Chennai