Innovation 2020
Organised by
Computer Engineering Department, Dr. D. Y. Patil School of Engineering, Lohegaon, Pune, Maharashtra, India in association with
International Journal of Scientific Research in Science, Engineering and Technology

Anomaly Based Intrusion Detection System Using Soft Computing and Data Mining Approach

Divyarani Babar, Dr. Pankaj Agarkar
Department of Computer Engineering, Dr. D. Y. Patil School of Engineering, Lohegoan, Savitribai Phule Pune University, Pune, Maharashtra, India

ABSTRACT

Data and application security is most essential in today environment due to highly resource utilization in network environment. Various network attacks detection and prevention techniques has already introduced by various researchers in many existing systems. Two identification of malicious behaviour from large traffic and take action against such request is the part of IDS. Various machine learning techniques also already developed to generate strong rules with different Optimization algorithms. But still IDS facing some issues like unknown attack detection accuracy, low accuracy for network attacks etc. However, cyber security threats are also growing as the contact points to the Internet are increasing. A significant security issue today is the intrusion detection system (IDS). A Network Intrusion Detection System (NIDS) helps system administrators to detect violations of network security within their operations. However, many problems arise when a robust and efficient NIDS is developed for unexpected and unforeseeable attacks. In this work, a deep learning based approach implement for effective and flexible NIDS. It is confirmed that the deep neural network is effective for NIDS through the performance test. System uses Recurrent Neural Network (RNN) which is supervised learning algorithm to detect known and unknown attacks into the both environments. Initially, Data pre-processing has done with Weka tool and define standard technique to eliminate unwanted records for attribute values. The proposed RNN algorithm works in both models for training and testing respectively. In first section we train the model with different network intrusion data sets (KDD CUP99, NSLKDD, ISCX, NB-15 etc.). Once a rule has created system deals with testing model an imbalance data generation environment. The partial implementation introduce proposed RNN provides better accuracy then other machine learning techniques. Additionally, we are evaluating and comparing different deep learning algorithms, namely RNN, CNN, DNN and PNN algorithm on cloud environment to detect intrusion in the network.

Keywords: Intrusion Detection System, Data Mining and IDS, KDD, WSN Trace Dataset

I. INTRODUCTION

The Intrusion detection system is defined as the system or software tool to detect unauthorized access to a network or computer system. Intrusion is a malicious, harmful entity which is responsible for network attack. This entity violates integrity, confidentiality and availability of a system resource. IDS is capable of detecting all types attack like malicious, harmful attack, vulnerability, data driven attacks and host based attacks.

Basically Intrusion Detection System (IDS) classified into two types- Host Based Intrusion Detection System (HIDS) and Network based Intrusion Detection System (NIDS). Today's system security foundation promisingly relies on Network intrusion detection Framework (NIDS)

[3,4,5]. NIDS gives security from known intrusion assaults. It is unrealistic to stop interruption assaults, so associations should be prepared to handle them. IDS is a cautious component whose main role is to keep work based on every conceivable assault on a framework. IDS is a procedure used to distinguish suspicious movement both at system and host level. Two principle methods of IDS used for accessing purpose are abnormality identification and abuse location. The oddity identification model depicts the typical conduct of a client to recognize the current client's irregular or unaccustomed activity.

II. METHODS AND MATERIAL

The proposed system works with deep learning approach. Program first collects examination data from various online and offline outlets. Once the data is collected through the program it applies pre-process and feature extraction. After the rules are created and stored into local database directory called as Background rules (BK Rules). Background rules are given as an input to the deep learning approach for the classification of sub attack. In this work, The RNN algorithm was applied to pre-processing distilled data to create a learning model, and the entire KDD Cup 99 dataset was used for testing. In the end, the accuracy, detection rate, and false alarm rate were determined to assess the detection efficiency of the RNN model.

The main objectives of this project are itemized as follows:

- The classification of attacks based on their characteristics is presented. Different components that make the detection of low-frequency attacks (like U2R and R2L, Worms, Shell Code etc.) hard to accomplish by machine learning strategies are examined and techniques are proposed for enhancing their detection rate.
- The discourse of different existing literature for intrusion detection is provided, featuring the key characteristics, the detection mechanism, feature selection is employed, attacks detection capability.
- The critical performance analysis of different intrusion detection techniques is provided with respect to their attack detection ability. The limitations and comparison with different methodologies are additionally talked about. Various suggestions are provided for enhancement in each category of techniques.
- Future headings of Deep learning are provided for intrusion detection applications.
- To generate strong and dynamic rules depending upon the real time behavior of the packet in training phase

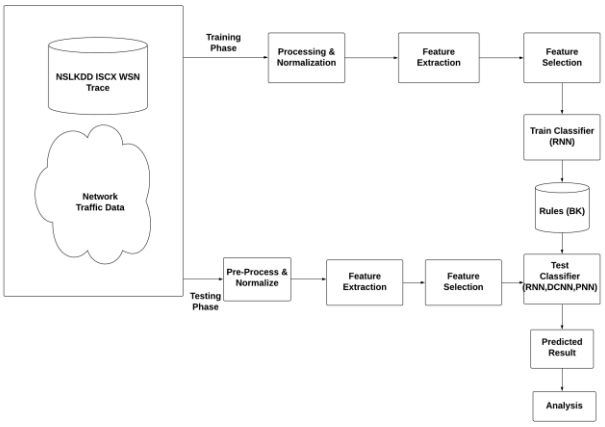

Figure 1: Proposed System Architecture

Training Phase:

Step 1: To generate the rules based on supervised learning algorithm we used synthetic dataset like KDDCup99, NSLKDD, ISCX and WSN Trace etc.

Step 2: Select features for each selected instances and execute the train classifier to generate the training rules.

Step 3: The result of training modules called as training rules or policies which has stored in repository those defined as Background Knowledge (BK).

System Testing Phase:

Step 1: System accumulate the network traffic data from network audit log data or NSLKDD

Step 2: Read each input packets from network environment and apply various machine learning as well as deep learning algorithm (RNN).

Step 3: RNN has apply to generate the runtime weight for each input packet and validate with the quality threshold.

Step 4: Classify the detected packet as master attack like DoS, PROBE, U2R, R2L, Network attacks etc), and finally also shows the subtype of attack for respective class.

Weight calculation using deep learning Algorithm (RNN)

Input: Train dataset which already store. Background knowledge by train classifier TD[], test dataset includes multiple pdf'sTestDb[], and desired threshold for validate the current weight.

Output: Hash_Map<class_label, sim_weight> all objects which having similarity weight larger than desired threshold.

Step 1: Read each test object using below function

$$testFeature(m) = \sum_{m=1}^{n} (.featureSet[A[i] \dots \dots .A[n] \leftarrow \text{TestDBLits})$$

Step 2: Extract each feature as a hot vector or input neuron from $testFeature(m)$ using below equation.

$$\text{Extracted\_FeatureSetx[t……n]} = \sum_{x=1}^{n}(t) \leftarrow testFeature\ (\text{m})$$

Extracted_FeatureSetx[t] contains the feature vector of respective domain

Step 3: extract each train objects using below function

$$trainFeature(m) = \sum_{m=1}^{n} (.featureSet[A[i] \dots \dots .A[n] \leftarrow \text{TrainDBList})$$

Step 4: extract features from each test set as best features for specific document object $testFeature(m)$ using below function.

$$\text{Extracted\_FeatureSetx[t……n]} = \sum_{x=1}^{n}(t) \leftarrow testFeature\ (\text{m})$$

Extracted_FeatureSetx[t] contains the feature vector of respective domain.

Step 5: Evaluate each test vector with entire train features and generate weight for respective instance

$$weight = calcSim\ (\text{FeatureSetx} \,||\, \sum_{i=1}^{n} \text{FeatureSety[y]})$$

Step 6: Return object [label] [weight]

- **MATHEMATICAL MODEL**

1) Let S be the system: Such that,

S= {Sys1,Sys2, Sys3, Sys4}

S1= Data preprocessing

S2= Feature Selection and Normalization

S3= Deep Learning Model

S4= Analysis

2) Let S1 be a data preprocessing phase:

S1= {TrainDB}

$$MI(x{:}c) = \sum_{k=0}^{n} (k=0)P(X=x, C=c).\log\left(P(X=x, C=c)\right)/\left(P(X=x)P(C=c)\right))$$

Where,

MI= preprocess Information

C= Class which can either be normal or anomaly

X= set of x vectors

3) Let S2 be a feature selection and normalization phase: S2= F1,F2, F3,..,Fn
F= All features in TrainDB
Policy for attribute selection:
Info = {protocol; service; duration; flag; srcbyte;dstbyte}
Where,
Info= Information feature selection

4) Let S3 be the deep learning model:
S3= {Test-Db, Packet(i), class}
Class= normal, anomaly
Packet= Network traffic packets
5) Let S4 be the analysis phase:
S4={Accuracy, Detection Rate}
Find accuracy of each classifier M.
Compare accuracy of each individual classifier with D.
Where,
D= deep learning model
Select best classifier model, i.e. M=D.
System basically consists of three phases like training phase,
testing phase and analysis phase. Here is the set dependency of the entire system.
System = {Train, Test, Analysis}
Train = {preprocess, feature extract, deep learning}
Test = {Pattern Match, Th, Weight, Subclass}
class = {Input →Bk-Rules→ Weight} {Normal; Attack} {sub attacks}
Analysis ={dos, probe, U2R, R2L, Normal, unknown

III.RESULTS AND DISCUSSION

The proposed, current machine learning algorithm and the deep learning algorithms were used in two different ways by the Project. We have also introduced computational research in base system which can recommend algorithms with KDDCUP99 data set and power-contributing architecture incorporated with deep learning algorithms with custom network audit dataset. The program measured the consistency of the description and the time complexity in the same setting. Figure 2 above demonstrates the classification performance of data collection by KDDCUP using the density-based approach of the machine learning algorithm program Figure 3 used to classify and predict the precision of the proposed system using different methods like RNN algorithm.

Existing System Results

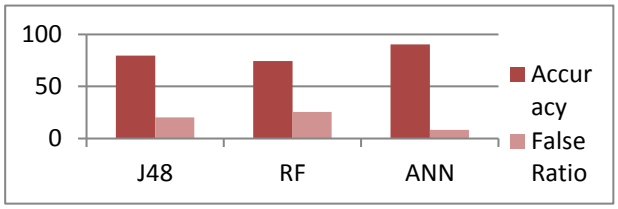

Figure 2:Detectionaccuracy for KDD :CUP99dataset using machine learning

The above figure 2 Shows accuracy of kddCup 99 results classification, with five different classes. Average software output is around the algorithm for the machine learning 88.50% for all classes.

Proposed Result

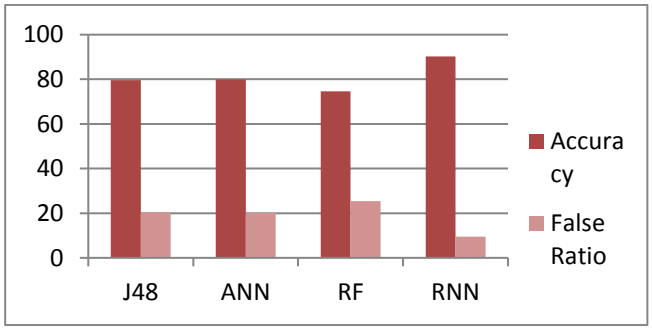

Figure 3 : Detection accuracy various network dataset using deep learning (RNN)

The above figure 3 Shows average efficiency of identification in various databases, of (n) different classes. The system's mean performance with the machine learning algorithm is around 95% for all (n) classes.

IV.CONCLUSION

In this work, we proposed a deep learning based RNN-IDS method to proposed effective intrusion detection system. We utilized the synthetic based intrusion dataset - NSL-KDD to evaluate anomaly detection accuracy. In future, we plan to implement an IDS using deep learning technique on cloud environment. Additionally, we Evaluate and compare different deep learning technique, namely. RNN, DNN, CNN and PNN on NSL-KDD dataset to detect intrusions in the network. The system basically works like machine learning as well as reinforcement algorithm to evaluate the unknown instances during the data testing. The effective rule system provides better classification and detection accuracy for classes. Various datasets are used for experiment analysis to evaluate the algorithm performance with multiple test and conclude we get result on satisfactory level.

V. REFERENCES

[1] Salo, Fadi, et al. "Data Mining Techniques in Intrusion Detection Systems: A Systematic Literature Review." IEEE Access 6 (2018): 56046-56058.

[2] Vinayakkumar, R., et al. "Deep Learning Approach for Intelligent Intrusion Detection System." IEEE Access 7 (2019): 41525-41550.

[3] Weiwei Chen, Fangang Kong, Feng Mei, GuiginYuan, Bo Li, “a novel unsupervised Anamoly detection Approach for Intrusion Detection System”, 2017 IEEE 3rd International Conference on big data security on cloud, May 16-18,2017, Zhejiang, China.

[4] Zaman, Marzia, and Chung-Horng Lung. "Evaluation of machine learning techniques for network intrusion detection."NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium.IEEE, 2018.

[5] Zhang, Hao, et al. "Real-time Distributed-Random-Forest-Based Network Intrusion Detection System Using Apache Spark." 2018 IEEE 37th International Performance Computing and Communications Conference (IPCCC). IEEE, 2018.

[6] Nathan Shone , Tran Nguyen Ngoc, Vu DinhPhai , and Qi Shi. Deep Learning Approach to Network Intrusion Detection”. IEEE Transactions on emerging topics in computational intelligence. VOL. 2, NO. 1, FEBRUARY 2018.

[7] Guangzhen Zhao, Cuixiao Zhang* and LijuanZheng.”Intrusion Detection using Deep Belief Network and Probabilistic Neural Networks”.IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC).2017.