Different Data Mining Techniques for Intrusion Detection System

Divyarani Babar[1], Dr. Pankaj Agarkar[2]

[1]Student, Dept. of Computer Engineering, Dr. D.Y. Patil School of Engineering Lohegaon Pune, India
[2]HOD, Dept. of Computer Engineering, Dr.D.Y. Patil School of Engineering Lohegaon Pune, India

---***---

Abstract - : *Intrusion detection system is a security system that serves the infrastructure as a protective layer. Intrusion Detection System (IDS) has emerged as one of the most complicated malware which encrypts the files once it enters into IDS the system. For the last couple of years, the rise in the number of IDS attacks with increasing severity, potency to cause high damage, ease of carrying out the attack has caused the conventional anti-malware techniques to pay attention and include advanced IDS detection mechanism. The IDS families, together characterized as crypto IDS, encode certain report sorts on inflamed structures and forces users to pay the ransom thru sure on line fee techniques to get the important thing for decryption. Hindering of IDS is essential in order to hoard the user's file. This paper proposes an approach by analyzing the program binaries both statically and dynamically to find specific properties for IDS. Our approach has identified such static IDS-specific properties along with the run-time properties, typically present during IDS execution that can be used for classification of IDS. The proposed system illustrates the approach of supervised learning which generate some Background Knowledge (BK) during the training phase and apply the same in testing phase.*

Key Words: **Data Mining, Intrusion Detection System, KDD, Network Attacks, IDS Attacks**

1. 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). Malware are of various kinds based on their behavior, properties and propagation technique. IDS is a type of malware with specific properties and objectives. It can lock the system or can encrypt the data as well. An attacker can demand a desired amount of money in return of decryption key. The advanced cryptographic algorithms are devised to protect valuable information and data, yet on the other side, these algorithms are being used to create malicious programs, specifically 'IDS'.

2. RELATED WORK

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

There are two different methods are available to detect malware in the system, "signature-based detection", and "anomaly-based detection" methods like supervised base machine learning systems. Each of the methods can be applied using one of the three techniques - Static approach, dynamic approach and hybrid approach. Figure 1 shows these detection methods and techniques with respect to IDS.

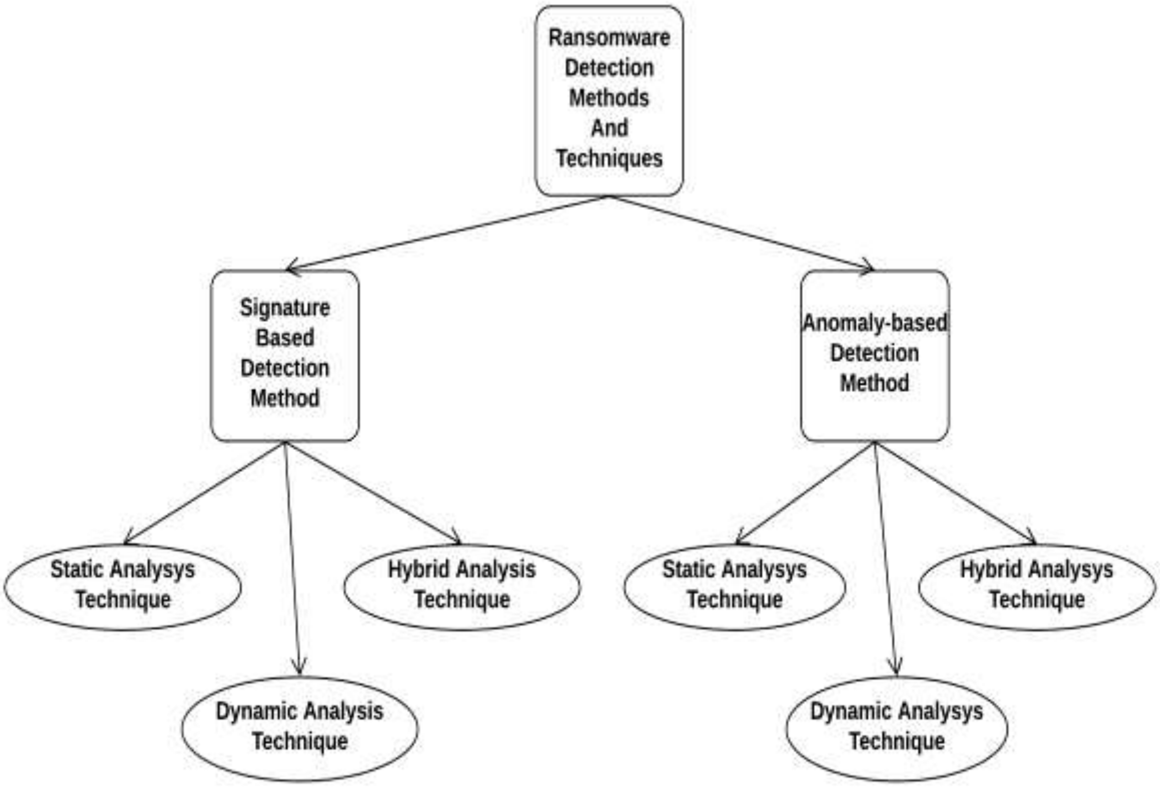
Ransomware Detection Methods And Techniques
Signature Based Detection Method
Anomaly-based Detection Method
Static Analysys Technique
Hybrid Analysis Technique
Dynamic Analysis Technique
Static Analysys Technique
Hybrid Analysys Technique
Dynamic Analysys Technique

Fig -1: IDS detection methods and techniques

2.1 IDS Detection Techniques

Three different strategies have been used in many existing approaches, a static method is the first version of supervised learning and dynamic is a bit advanced than. The hybrid method is basically a combination of static as well as dynamic method which works like reinforcement learning approach, each of detection methods is explained below.

Static analysis: In this technique, the syntax or structural features of malware are used. This technique is used for detection before malware is run. In IDS systems when packets arrive on victim's port, the system captures it and store in own buffer class. The buffer class does not allow entry to such packets in machine or device. During this phase, the system generates hash values for the received packet and maps the features similarity with database signatures and defines the current file or packet is malicious or normal.

Dynamic analysis: In this technique, runtime information of malware is used. This technique is used for detection while malware is running or afterward. During the execution it generates the runtime hash of current metadata and evaluates with the database pattern, the similarity weight of current hash verified with quality threshold values and assign the class label to the current test data connection.

Hybrid analysis: This technique consists of the use of the features of two techniques which is already mentioned. This technique is used for detection before, during, and after the malware is run. It has superior benefits over other approach. The proposed system works with similar detection approach which is already used in some existing machine learning base IDS.

3. ARCHITECTURE

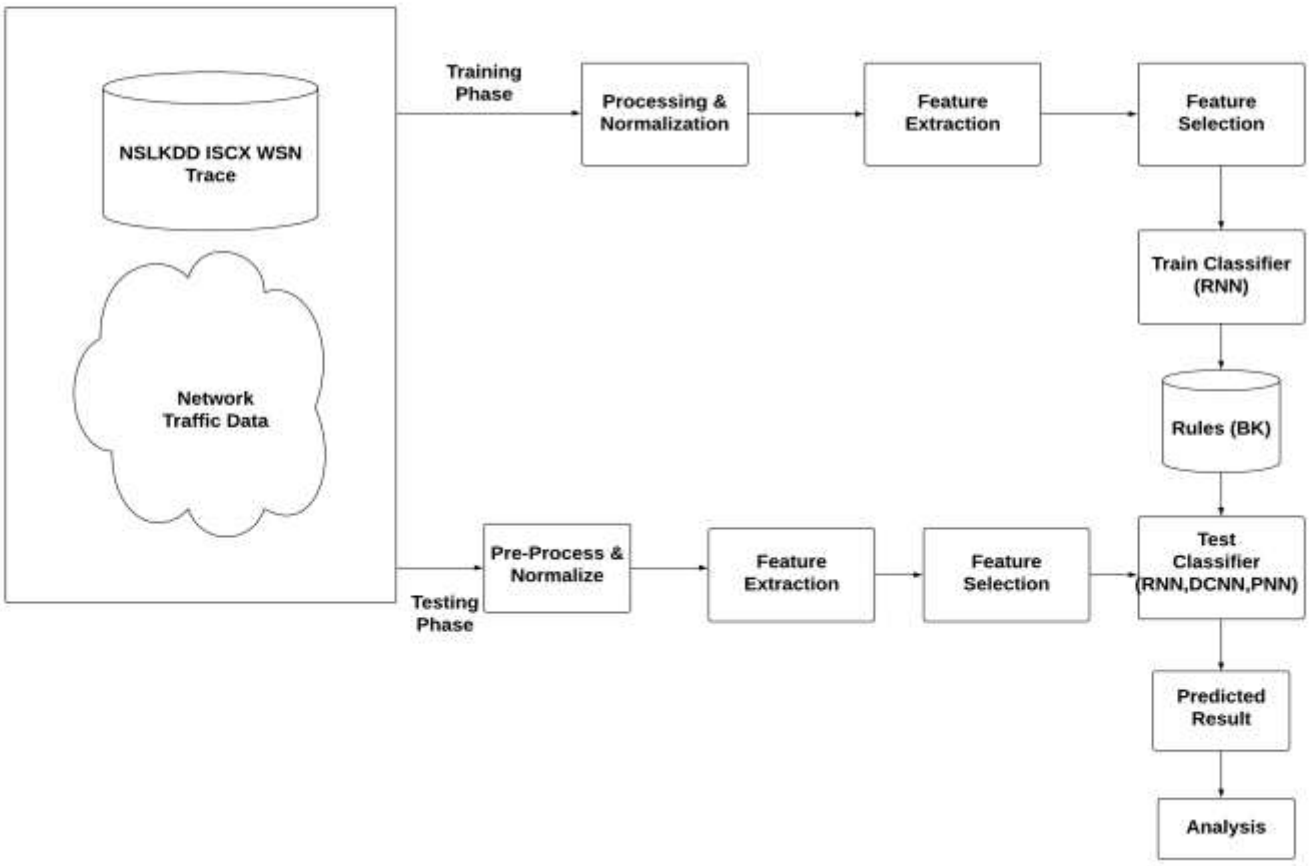

Fig -2: Proposed System architecture

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, and Network attacks etc), and finally also shows the subtype of attack for respective class.

4. ALGORITHMS

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

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

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

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

Extracted_FeatureSetx[t] contains the feature vector of respective domain

Step 3: extract each train objects using below function

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

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

Extracted_FeatureSetx[t……n] $=\sum_{x=1}^{n}(t) \leftarrow testFeature$ (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]

5. MATH

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}

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

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}

6. 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

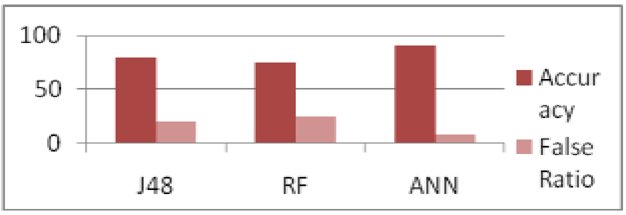

Chart -1: Detection accuracy for KDD

The above chart 1 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

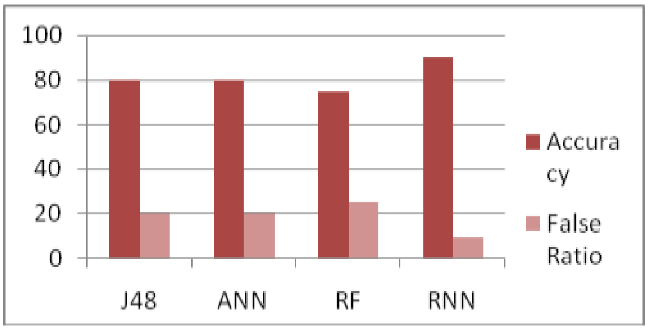

Chart -2: Detection accuracy for various network datasets

The above chart 2 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.

7. 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.

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] 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.

[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] Zhou, Yiyun, et al. "Deep learning approach for cyberattack detection." IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS).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.