Wireless Network Intrusion Detection using

K-Means Clustering Algorithm

V. Akshay – 2015503005

J. Timothy Jones Thomas – 2015503561

S. Vikraman – 2015503564

***Abstract -*This project is to create an Intrusion Detection System (IDS). An Intrusion Detection System is an application that monitors the network for malicious activities and unauthorized access to device information as well as personal data. The Intrusion Detection System is based on the k-means clustering algorithm. This algorithm partitions n observations into k clusters, in which each observation belongs to the cluster with the nearest mean, thus serving as a prototype for that cluster. The initial data set is partitioned into such clusters and by making use of these, the cluster to which the test data belongs can be predicted. Based on this predicted cluster, the application notifies whether the test data is normal or suspicious. If a new or unknown type of attack takes place, the application will find the data to be deviant from the rest and will mark it as suspicious. The initial training data space is obtained from the KDDCUP99 dataset.**

***Keywords – intrusion detection system; network security; k-means clustering; packet sniffing;***

I. INTRODUCTION

With growing susceptibility to attacks, user data is prone to huge risks. Hence, network security is of paramount importance. Valuable resources having network access should be permanently protected from all attempts to destroy, expose, alter, disable, steal or gain unauthorized access and/or usage. Resources confidentiality, integrity and availability have to remain intact. Intrusion detection system (IDS) is a system specially designed to detect such malicious attempts. As traditional IDS's are mainly signature-based, detecting only known attacks, their biggest problem is the inability to detect new or variant attacks. One topic that intuitively stands out as a potential solution for solving this problem is k-means clustering.

II. RELATED WORKS

Neural Network (NN) is the most popular AI algorithm used for intrusion detection compared to other algorithms. However, training these networks takes a lot of time to achieve a reasonable level of performance, and also their adaptability is unsatisfactory [1]. Recent IDSs based on Naïve Bayes and Decision Trees seem promising, with better accuracy and performance. Genetic Algorithms and Support Vector Machines (SVM) are also being used, though it has been stipulated that the accuracy of SVMs are on the lower side [1].

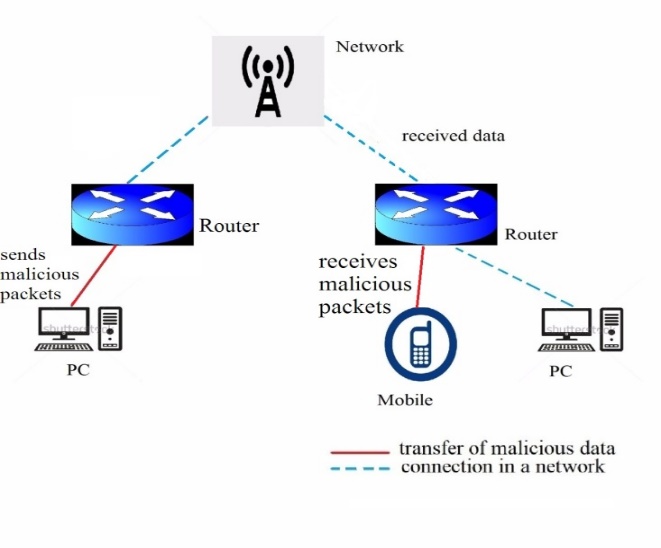
Bisyron Wahyudi Masduki [2] used the following machine learning algorithms - KNN, SVM and Dempster Shafer theory. Firstly, a few features from KDDCUP are selected as training data. KNN and SVM are used to classify attack data and the output of those two different methods is combined with Dempster Shafer Theory. The performance of the classification process is good in overall, but the results are not optimal to detect R2L and U2R attacks categories.

Yanjie Zhaossss [3] studied classifier model on a training dataset which has very different class distribution. They proposed PN rule, a two-stage general-to-specific framework of learning a rule-based model. This method can detect only 10.7% of attacks in the attack class R2L despite a lot of false alarms generated. A real shortage of this method is that the rule is determined automatically, which makes it dependent on the dataset.

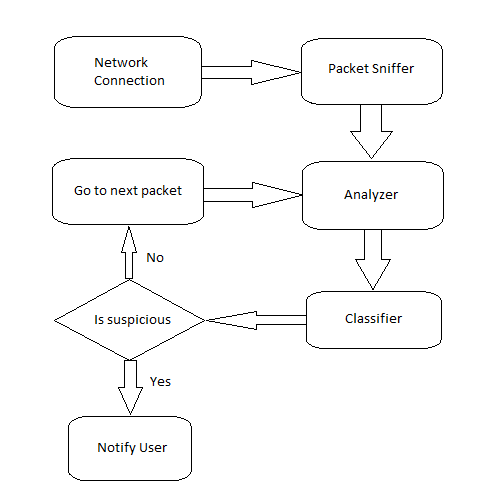
Lofti Mhamdi [4] used Multilayer Perceptron(MLP) which is a supervised learning algorithm based on the feed-forward neural network with one or more layers between input and output layer. Tuan A Tang [4] used MLP for anomaly detection, where the proposed model is a single hidden layer neural network. However, the proposed classifier achieved only 7.32% detection and 2.5% false alarm rates for the R2L attacks.

III. PROPOSED WORK

Initially, the knowledge base is created by using pre-existing datasets (KDDCUP99) to form clusters. This clustering is done by the k-means clustering algorithm. Each cluster represents the different types of network access.

Fig 1: Topology of a network with transfer of malicious packets

A sample network is shown in Fig. 1, where the device that sends the malicious packets and the device that receives them, are on the same network. The receiver has an Intrusion Detection System, which scans the packets that are received, fetches the required parameters, and sends it to the classifier. If the classifier deems the packet as suspicious, the user is notified. Else, the process is repeated for the next packet. The architecture for the Intrusion Detection System is represented in Fig. 2.

Fig. 2: Architecture for the intrusion detection system

IV. ALGORITHM AND MATHEMATICS INVOLVED

*k*-means clustering is used to model the knowledge base. Each observation belongs to the cluster with the nearest mean. These clusters are plotted on a plane. If an observation is plotted near a cluster, it is assumed to belong to that cluster. Hence, these clusters should be placed in such way that there is no ambiguity in association. To achieve this, it is better to place the clusters as far as possible from each other.

We develop cluster models based on the training data set for multiple values of k, which represents the number of clusters in the model. The k value for which the variance of clusters is minimum is chosen as the best model. This is equivalent to minimizing the pair-wise squared deviations of points in the same cluster. This is also equivalent to maximizing the squared deviations between points in different clusters. This is done by using the training dataset once again.

Each Observation is a tuple with *n* values, and the *i*th value in the tuple is represented as *obs[i]*. The deviation or distance between two observations in the cluster model is given as,

To find which cluster a given observation belongs to, the model calculates the deviations between the cluster centroids and the given observation. The observation is said to belong to the cluster to which it is nearest i.e., the deviation is minimum. This is the basis for classification.

By finding the deviation between each record in the training dataset and the centroid of the cluster to which it belongs, we get the variance for that record. Each record is a tuple consisting of *n* values.

The mean of the variances, for all *N* records of the training set, is taken as the clustering score.

Once the clustering scores are calculated for multiple *k* values, the model with the lowest score is taken as the best model. This model is used for classifying the test data. The cluster to which most of the normal data is mapped, is found. If the test data is mapped to any other cluster, it is marked as suspicious. Else, it is marked as normal.

While evaluating IDS, for every possible test value there are two kinds of error: false positive (FP) and false negative (FN). FP occurs when an event is predicted as normal but it is, in fact, intrusive, while FN occurs when a normal event occurs without being recognized as one. On the other hand, true positive (TP) measures the proportion of actual positives which are correctly identified as such, while true negative (TN) measures the proportion of negatives which are correctly identified as such. The performance of the classifier can be quantified using the detection rate (DR) and overall accuracy (OA) measures [1]. DR shows the percentage of the true intrusions that have been successfully detected:

OA is calculated as the total number of correctly classified intrusions divided by the total number of observations:

V. IMPLEMENTATION

The Intrusion Detection System is developed using Python programming language. The Apache Spark (pyspark) Machine Learning library(mllib) is imported, which contains the implementation of the k-means clustering algorithm. The following inbuilt functions were used: KMeans.train(), KMeansModel.predict(). The train() function returns a KMeansModel object which is used for prediction. The predict() function returns the predicted cluster index.

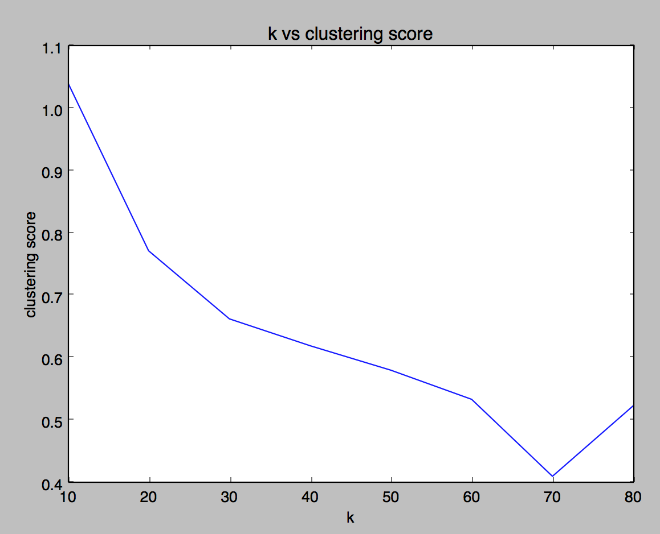
VI. RESULTS AND PERFORMANCE ANALYSIS

The Clustering scores for different values of k are shown in Table 1 and Graph 1. From the values, it can be deduced that k = 70 gives the lowest score. Hence it is chosen for prediction, though higher values of k can give lower scores.

Table 1: Clustering Scores for different values of k

|  |  |
| --- | --- |
| **Value of k** | **Clustering Score** |
| 10 | 1.041418 |
| 20 | 0.771934 |
| 30 | 0.662832 |
| 40 | 0.619724 |
| 50 | 0.580507 |
| 60 | 0.533794 |
| 70 | 0.410462 |
| 80 | 0.523469 |

Graph 1: Clustering Scores



The test data set is given as input and the Detection Rate and Overall Accuracy are calculated. Table 2 displays the results of running the test data set.

Table 2: Results of Prediction

|  |  |
| --- | --- |
| **Type of Prediction** | **Number of Observations** |
| True Positive | 50003 |
| False Positive | 9804 |
| True Negatives | 240632 |
| False Positives | 10590 |

Based on the Table 2, the Detection rate (DR) is calculated to be 82.52% and the Overall Accuracy is 93.44%.

VII. CONCLUSION

Data security is a major concern for everyone. In this paper, we discussed the k-means clustering algorithm conceptually, for classifying the test values as malicious or normal. This algorithm can be integrated into an application in a real-time system to monitor the network. In this way, network security can be ensured.

VIII. REFERENCES

1. M. Stampar and K. Fertalj, “Artificial Intelligence in Network Intrusion Detection”, 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), DOI: 10.1109/MIPRO.2015.7160479, pp. 1318, 2015
2. Bisyron Wahyudi Masduki, Kalamullah Ramli, “Improving Intrusion Detection System Detection Accuracy and Reducing Learning Time by Combining Selected Features Selection and Parameters Optimization”, 2016 6th IEEE International Conference on Control System, Computing and Engineering (ICCSCE), DOI: 10.1109/ICCSCE.2016.7893606, pp. 397, 2016
3. Yanjie Zhaossss, “Network Intrusion Detection System Model Based on Data Mining”, 2016 17th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), DOI: 10.1109/SNPD.2016.7515894, pp. 978, 2016
4. Tuan A Tang, Lotfi Mhamdi, Des McLernon, Syed Ali Raza Zaidi and Mounir Ghogho, “Deep Learning Approach for Network Intrusion Detection in Software Defined Networking”, 2016 International Conference on Wireless Networks and Mobile Communications (WINCOM), DOI: 10.1109/WINCOM.2016.7777224, pp. 1, 2016
5. Sonu Duhan, Padmavati Khandnor, “Intrusion Detection System in wireless sensor networks: A comprehensive review”, 2016 International Conference on Electrical, Electronics, and Optimization Techniques(ICEEOT), DOI: 10.1109/ICEEOT.2016.7755187, pp. 2707, 2016
6. Carlos Garcia Corderob, Emmanouil Vasilomanolakisa,b, Nikolay Milanovc, Christian Kochc, David Hausheerc, Max Muhlhauser, “ID2T: a DIY Dataset Creation Toolkit for Intrusion Detection Systems”, 2015 IEEE Conference on Communications and Network Security (CNS), DOI: 10.1109/CNS.2015.7346912, pp. 739, 2015
7. Dikshant Gupta, Suhani Singhal, Shamita Malik, Archana Singh, “Network Intrusion Detection System using various data mining techniques”, International Conference on Research Advances in Integrated Navigation Systems (RAINS), DOI: 10.1109/RAINS.2016.7764418, pp. 16, 2016