# Anomaly based Intrusion Detection System

Janardan Bhatta[1],Kushal Gajurel[2], Santosh Nepal[3] , Saurav Pandey[4], Shekhar Koirala[5]

janardan.bhatta@ioe.edu.np[1], gajurelkushal1994@gmail.com[2], nep.santo001@gmail.com[3], pysaurav@gmail.com[4], shekharkoirala4@gmail.com,[5]

## ABSTRACT

*With the rapid development in the computer network systems, the challenge to ensure network security is also increasing. To maintain the higher level of security in network system, this paper presents 'Anomaly based intrusion detection system'; an approach to detect the anomalous activities in the network and overcome the weakness of signature based detection system, by using predictive models capable of distinguishing between intrusions or attacks connections and normal connections. This system is based on machine learning, which incorporates different machine learning algorithms such as Naive Bayes classifier, KNN algorithm, Decision tree classifier and validating the outcomes between them. In order to make the model non-static, a similar sklearn model is built using Azure cloud platform. Raspberry PI is then used to tap Network log using tcpdump packet analyzer and is fed to our trained system in cloud.*

*Keywords: Anomaly Detection, KNN, Naive Bayes, Decision tree,KDD, tcpdump.,intrusion detection system*

## I. INTRODUCTION

An intrusion detection system deals with monitoring network traffic and suspicious activities, generating alerts and in many cases, respond to anomalous or malicious traffic by blocking the user or source IP address. It, however, doesn't include network loggers, anti-viruses,cryptographic systems and firewalls. In signature based network IDS,a collection of signatures is maintained, which characterizes the profile of a known security threat. Signature based designs have low false positives,and are effective in cases of known threats and established signatures but are ineffective against unknown threats. On the other hand, anomaly based NIDS can be used to detect and contain security violations before they propagate and cause any damage.While signature based systems are reactive and require manual support, anomaly based system are autonomous, proactive and ensure security without manual interference [1].

## II. LITERATURE REVIEW

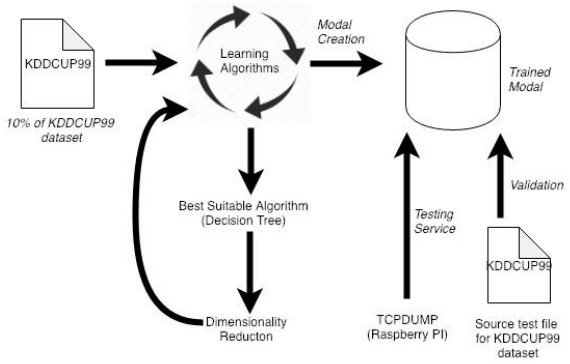
IDS was first introduced by Denning in her seminal work on the host-based IDS system in 1987. To capture normal activity, a combination of statistical metrics and profiles were used.[2]. In [3],Geetha Ramani et al. used a statistical method for analyzing the KDD 99 dataset where important features were identified by studying the internal dependencies between features. Statistical Packet Anomaly Detection Engine (SPADE), a plug-in for the open source IDS Snort,and requires minimal resource to inspect recorded data for anomalous behavior based on a computed score.

Login Anomaly Detection (LAD),built by Psionic Technologies implemented User Entity Behavior Analysis (UEBA) in their HostSentry program.It uses flow-based anomaly detection which characterizes and tracks network activities to differentiate abnormal network behavior from normal.

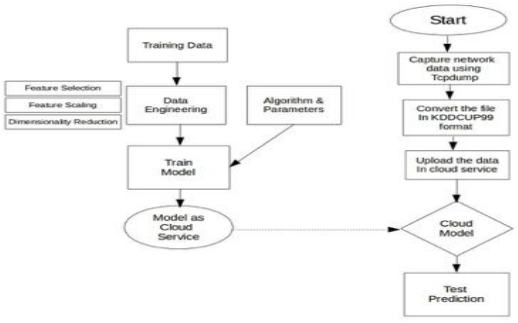
As per research paper [4], an anomaly based NIDS is capable of detecting high volume traffic flows, flash crows, load imbalance in the network, sudden changes in demand of a port usage, sudden surge of traffic from/to a specific host.IDS can also recognize, with a

certain false positive probability, new attacks and abnormal patterns in the network traffic, whose signatures are not yet generated. Almost all NIDS systems require a constant human supervision which slows down the detection and the associated actions[5].

## III. SYSTEM DESIGN




The input of the system is 10% of KDDCUP99 Dataset, used in repeated process of model creation using different algorithm of Python scikit-learn library and final best possible model is created.Equivalent machine learning model is built using Microsoft Azure Machine learning platform without features change. Azure enable us to create a web API and API key which we can be used in any web based application. Similarly, a Raspberry pi3 is connected to a network generate network traffic log file using Tcpdump package available




in Linux platform. The generated log file undergoes data preprocessing steps and uploaded in excel online or we can test using only azure machine. For immediate evaluation in our model, “corrected” dataset, which is defined test dataset for KDDCUP99 dataset, is used. The dataset is also uploaded in the the cloud services and the output is predicted.

**Algorithms used :**

**k-Nearest Neighbors Algorithm:**

The k-Nearest Neighbors (k-NN) algorithm is a non-parametric classification and regression. Input consist of the examples in the feature space.In k-NN classification, the output is a class membership.In k-NN regression, the output is the property value for the object.[6]

**parameters**: n_neighbors=5, weights='uniform', algorithm='auto', leaf_size=30, p=2, metric='minkowski', metric_params=None, n_jobs=1,

**Logistic Regression**

The binary logistic model is used to estimate the probability of a binary response based on one or more independent features. The presence of a risk factor increases the probability of a given outcome by a specific percentage. The logistic regression can be understood simply as finding the β parameters that best fit: $y = \{1\ \beta 0 + \beta 1x + \varepsilon > 0, 0$ else where ε is an error distributed by the standard logistic distribution. The associated latent variable is $y' = \beta 0 + \beta 1x + \varepsilon$ . The β parameters cannot be expressed by any direct formula of the y and x values in the observed data. The y values are to be found by an iterative search process, that finds the maximum of a complicated likelihood expression,which is a function of all of the observed y and x values.

**parameters**:

(penalty='l2',dual=False,tol=0.0001,C=1.0,fit_intercept=True,intercept_scaling=1,class_weight=None,random_state=None,solver='liblinear',max_iter=100,multi_class='ovr',verbose=0, warm_start=False, n_jobs=1)

**Random Forest**

It is an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and

outputting the class that is the mode of the classification or mean prediction of the individual tress. Random decision forests correct for decision trees' habit of overfitting to their training set.

**Decision Tree**

Decision tree learning is a predictive model to go from observations around an item to conclusions about the item's target value. It is predictive modelling approaches used in statistics, data mining and machine learning. Decision trees where the target variable can take continuous values are called regression trees. Decision tree learning is the construction of a decision tree from class-labeled training tuples. A decision tree is a flow-chart-like structure, where each internal (non-leaf) node denotes a test on an attribute, each branch represents the outcome of a test, and each leaf (or terminal) node holds a class label. The topmost node in a tree is the root node.

The attribute selection measure taking into account the discriminative power of each attribute over classes in order to choose the 'best' one as the root of the (sub) decision tree. In other words, this measure should consider the ability of each attribute Ak to determine training objects' classes. We mention the gain ratio, is based on the Shannon entropy, where for an attribute Ak and a set of objects T, it is defined as follows:

$$Gain(T, Ak) = Info(T) - Info_{A_k}(T) \quad (1)$$

$$Info(T) = -\sum_{i=1}^{n} \frac{freq(c_i, T)}{|T|} log_2 \frac{freq(c_i, T)}{|T|} \quad (2)$$

$$Info_{A_k}(T) = \sum_{a_k \epsilon D(A_k)} \frac{|T_{a_k}^{A_k}|}{|T|} Info(T_{a_k}^{A_k}) \quad (3)$$

and freq($c_i$, T) denotes the number of objects in the set T belonging to the class $c_i$ . Then, Split Info($A_k$) is defined as the information content of the attribute Ak itself.

$$Split \quad Info(T, A_k) = -\sum_{a_k \epsilon D(A_k)} \frac{|T_{a_k}^{A_k}|}{|T|} log_2 \frac{|T_{a_k}^{A_k}|}{|T|} \quad (4)$$

So, the gain ratio is the information gain calibrated by Split Info.

$$Gain\, ratio(T, A_k) = \frac{Gain(T, A_k)}{Split\, Info(A_k)} \quad (5)$$

**parameters:**(criterion='gini',splitter='best',max_depth= None,min_samples_split=2,min_samples_leaf=1,min_ weight_fraction_leaf=0.0,max_features=None,random_ state=None,max_leaf_nodes=None,min_impurity_split =1e-07, class_weight=None, presort=False)

**Gaussian Naïve Bayes (NB)**

Naïve Bayes classifiers is a simple probabilistic classifiers implementing Baye's theorem with strong independence assumptions between the features.While dealing with continuous data, it is assumed that the continuous values associated with each class are distributed according to a Gaussian distribution. If the training data contains a continuous attribute x, We first segment the data by class, and then compute the mean and variance of x in each class.

Naive Bayes is expressed by following formula

$$P(c_i|A) = \frac{P(A|c_i) \cdot P(c_i)}{P(A)} \quad (6)$$

where $c_i$ is a possible value in the session class and A is the total evidence on attributes nodes.

**parameters**: priors

**ID3**

Iterative Dichotomiser 3, is an algorithm used to generate a decision tree from a dataset. It begins with the original set S as the root node. On each iteration of the algorithm, it iterates through every unused attribute of the set and calculates the information gain of that attribute. It then selects the attribute which has the largest information gain value. The set is then split by

the selected attributed to produce subsets of the data. The algorithm continues to recurse on each subset, considering only attributes never selected before.Throughout the algorithm, the decision tree is constructed with each non-terminal node representing the selected attribute on which the data was split, and terminal nodes representing the class label of the final subset of this branch.

**parameters**: (criterion='gini', splitter='entropy', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_split=1e-07, class_weight=None, presort=False)

**Linear SVC**

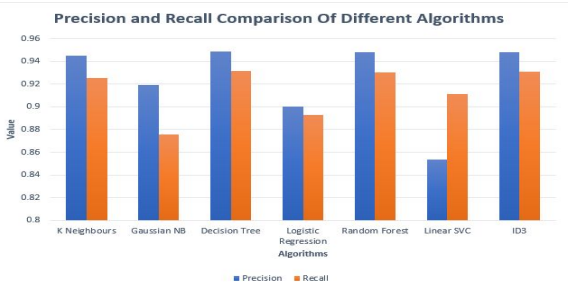
In machine learning, support vector machines (SVMs) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification implicitly mapping their inputs into high-dimensional feature spaces. When data are not labeled, supervised learning is not possible, and an unsupervised learning approach is required, which attempts to find natural clustering of the data to groups, and then map new data to these formed groups. The clustering algorithm which provides an improvement to the support vector machines is called support vector clustering (SVC) and is often used in industrial applications either when data are not labeled or when only some data are labeled as a preprocessing for a classification pass.

**parameters**:(penalty='l2',loss='squared_hinge',dual=True,tol=0.0001,C=1.0,multi_class='ovr', fit_intercept=True, intercept_scaling=1, class_weight=None, verbose=0, random_state=None, max_iter=1000)

The parameter used in our system is tuned. Parameter needed to be adjusted for best possible result.

## IV. RESULTS

Some statistical analysis is done in the input data which leaded to do data selection. Various algorithm is used and its' measure is calculated. Even Decision Tree , Random Forest and ID3 shows similar stats. The Time taken to train and test data is minimal in Decision Tree.

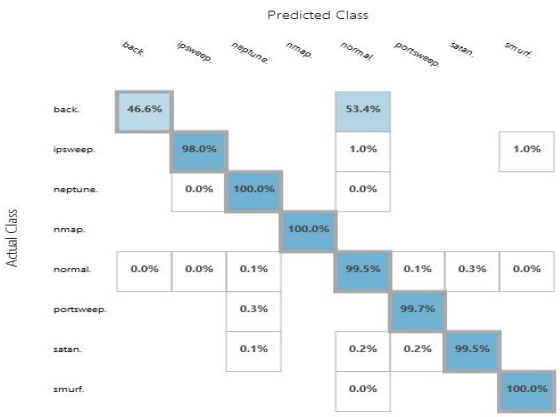



Using the best performance algorithm, dimensionality reduction from 42 column of feature to 22 feature was accomplished. Here dimension reduction is done by Permutation feature importance, works by randomly changing the values of each feature column, one column at a time, and then evaluating the model.

Permutation feature importance does not measure the association between a feature and a target value, but instead captures how much influence each feature has on predictions from the model.

| Before dataset reduction : | After dataset reductio: |
|---|---|
| Overall accuracy::0.996764 | Accuracy :0.9316143 |
| Overall precision::0.963953 | Precision : 0.9486956 |
| Overall recall:0.929219 | Recall :0.9316114 |

**Confusion matrix:**

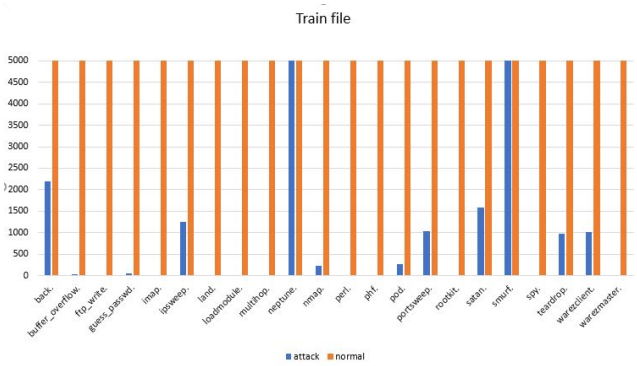



## V. CONCLUSION AND FUTURE ENHANCEMENT

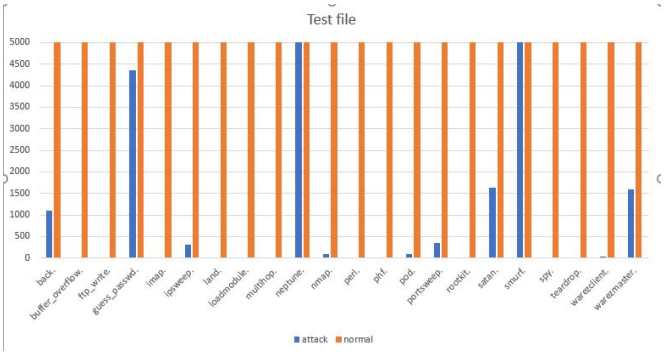
*Implementing machine learning in network security is a precarious matter and to detect attack d data of attack on network is required, which is difficult to obtain. So using KDDcup99 data, we tested the feasibility of present machine learning algorithms to understand the network data. Feature engineering was practised and the data obtained from tcpdump packet analyzer was parsed into the fields as that of kddcup99 data set. The algorithms, set to different parameters resulted in diverse performance parameters. For the given set of data, we observed that decision tree classifier worked the best than other algorithms. The model was deployed in cloud services and monitored using IOT devices.*

**Appendix: Data Selection**

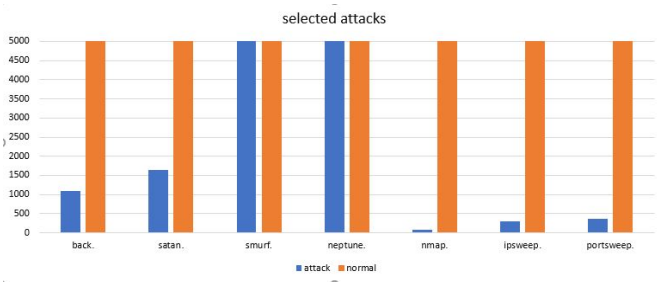
Details of Train File




Details of Test File




Details of Selected Attack




REFERENCES

[1]"Survey of current network intrusion detection techniques,".[Online].Available: http://www.cse.wustl.edu/~jain/cse571-07/ftp/ids/. Accessed: Feb. 18, 2017.

[2].D.E.Denning,“An Intrusion-Detection Model,”IEEE Transactions on Software Engineering, vol. 13, no. 2, pp. 222–232,1987

[3]GeethA.Srivastava,[Online].Available:https://www.raspberrypi.org/magpi/raspberry-pi-3-specs-benchmarks/. Accessed:Feb. 19, 2017.a Ramani R, S.SivaS athya, Siva selviK, "Discriminant Analysis based Feature Selection in KDD Intrusion Dataset,",International Journal of Computer Application VoI.31,No.ll, 2011

[4] A. Lakhina, et al., "Mining Anomalies Using Traffic Feature Distributions," Proc. ACM SIGCOMM 2005.
www.sigcomm.org/sigcomm2005/paper-LakCro.pdf

[5] Cisco IOS IPS Deployment Guide. www.cisco.com

[6]Varuna, S., & Natesan, P. (2015).An integration of k-means clustering and naïve bayes classifier for Intrusion Detection. *2015 3rd International Conference on Signal Processing, Communication*

*and Networking (ICSCN)*. Doi: 10.1109/icscn.2015.7219835