VIT

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

BCI 3002 : Disaster Recovery and Business Continuity Management (DRBCM)

Slot: A1+TA1

Review 1

TITLE : PREVENTION OF ATTACKS USING ONE CLASS CLASSIFICATION AND AUTO ENCODERS

6th October 2021

Team Leader:
Rohan Allen 18BCI0247

Team Members:
Rakshith Sachdev 18BCI0109
Harshita Pundir 18BCI0192

AIM AND SCOPE:

The aim of this project is to train our system to differentiate between bad network traffic which might contain some virus, malware to the system, or any other type and normal network using machine learning. The model has also been trained to predict false data and henceforth prevent the installation of any particular software during the risk of an attack, which results in an increased cost. The main key objective is to provide maximum accurate results by using / one class-based modeling approach and reducing the processing time significantly.

ABSTRACT AND PROBLEM STATEMENT:

With today's day and age rapidly becoming digital, the network and end point devices become a target for attacks and exploitation; thus, the systems have long been associated with issues related to security. Therefore, making systems secure and safe is of extreme importance. As per the 2020 Unit 42 Threat Report, practically all traffic is decoded, implying that the majority of classified and individual user information in the network is highly powerless against cyber attacks. Network security is utilized to delay unintentional harm which can be done to the network's private information, its users, or its devices. The main aim of network security is to secure the network running and for every single authentic client.

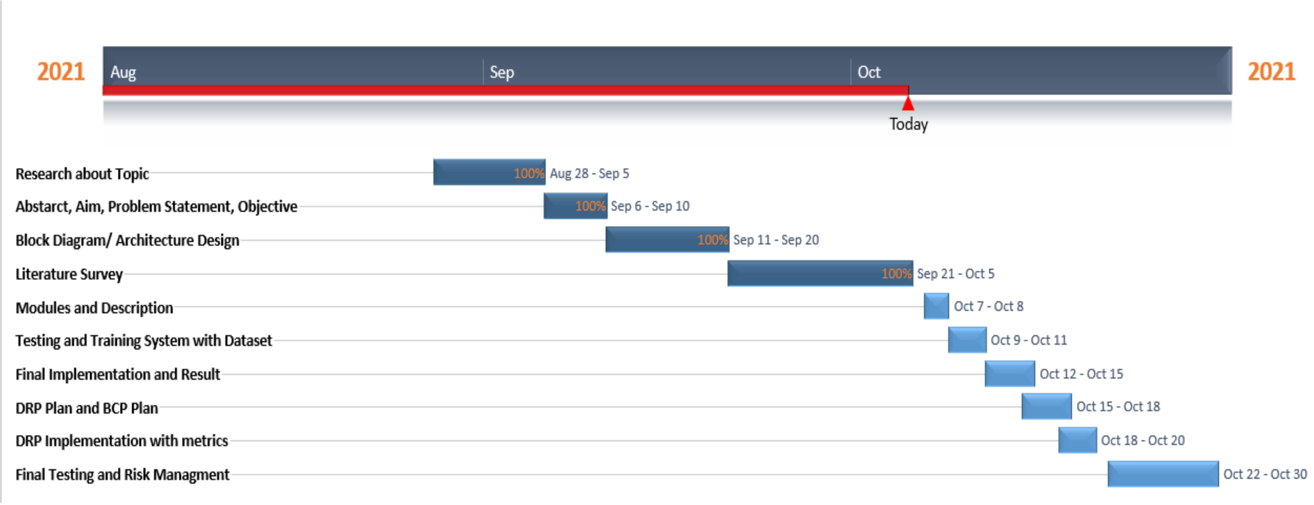
In this project, we represent how one-class classifiers are prepared to utilize generous information to recognize ordinary and dangerous traffic redirected to an end point device. In this venture, the framework is prepared to utilize unsupervised / one-class-based demonstrating approaches by which the framework would comprehend the issues that we would confront day by day, and the preparation will be useful for what's to come. After the preparation of the framework, the framework can be utilized in reality to confront ongoing, new difficulties and by gaining from the past experiences it can develop as indicated by the users' issues, weaknesses, dangers, and conditions

OBJECTIVES:

The objective of this project is to use machine learning to teach our system to distinguish between malicious network traffic, which could contain a virus or malware, and regular network data. The model has also been taught to detect and avoid fake data and the deployment of any specific software while there is a threat of an attack, which results in dramatically reducing the financial stress on an organisation and prevents tarnishing their reputation . The major goal is to offer the most accurate findings possible by utilizing methods such as unsupervised learning/one-class-based modeling, thereby lowering processing time substantially.

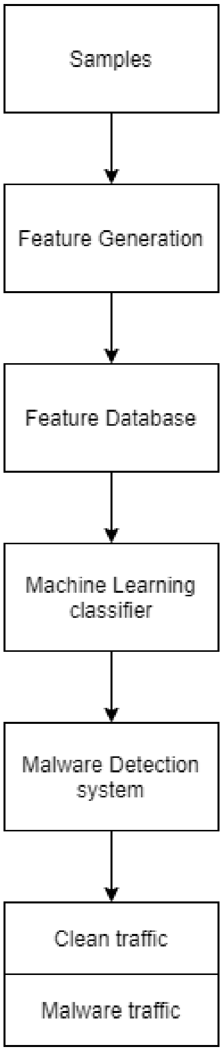
In this model, we are using one-class classifications and autoencoders so that the system can detect bad traffic more accurately. With the help of this model, we can predict false data, and therefore, we can prevent the installation of software at the time of any risks to decrease the cost.

SCHEDULE DIAGRAM OF THE SYSTEM:



BLOCK DIAGRAM OF THE SYSTEM / ARCHITECTURE OF THE SYSTEM:

1 High Level Design



Samples are the input datasets that are entered into the model. The input datasets will include both uncorrupted and corrupted data so as to give visible results in the end when the traffic is separated into good and malware.

A feature is a novel characteristic property of the process being viewed. Data collection and processing can take a lot of time and cost to undergo. In cases of regression, pattern recognition and classification, finding and segregating which features are important to be considered is vital especially in scenarios including datasets of huge sizes in terms of features and instances. Thus, feature generation can help in sorting out this issue.

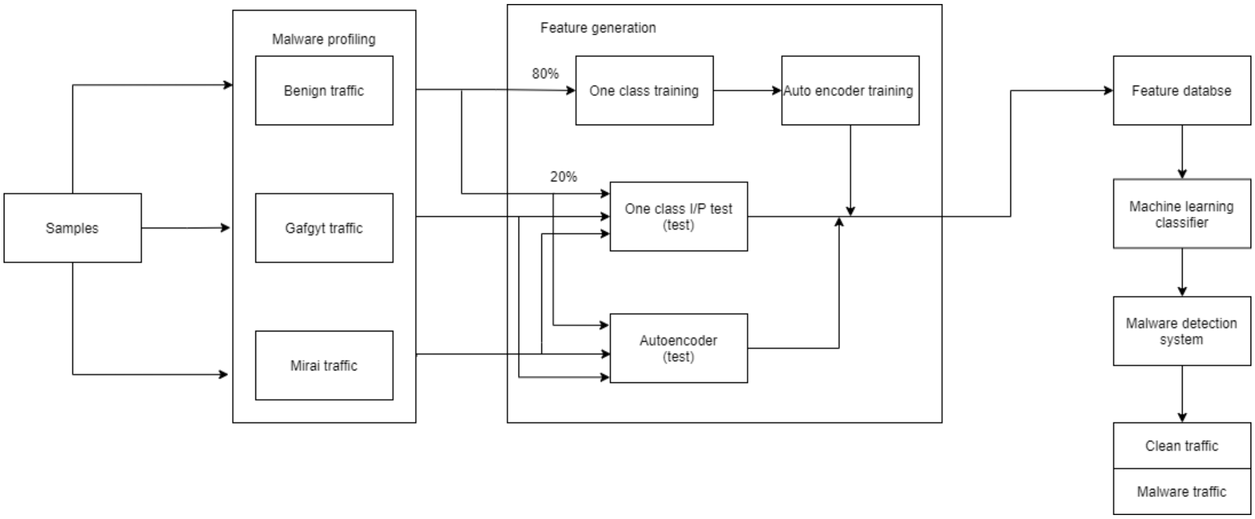
Feature generation helps invent new features from one or more prevailing features, for further use in statistical analysis. This gives fresh information regarding the dataset available at the time of building the model which can potentially lead to a model with higher accuracy. Feature database stores the features involved with the dataset for the model like a data warehouse.

The machine learning classifier is an algorithm that assists in ordering or categorizing the given input data inevitably into different categories. Here, the classifier will sort the traffic given as input into clean and malware traffic. Unsupervised machine learning classifiers follow anomalies and pattern structures or recognition in the data based on the unlabelled datasets loaded into the model

Malware detection system will ensure that the good traffic consists of only benign data with the malware traffic which contains malicious data and attacks removed from it.

Finally, the output is the input traffic given to this model is divided into both malicious and benign traffic, with a high percentage of accuracy.

2 Low Level Design



There are three types of datasets given as input - benign traffic containing 40,395 records, Mirai traffic containing 652,100 records, and Gafgyt traffic containing 316,650 records. Benign traffic is the good traffic containing clean data whereas Gafgyt and Mirai traffic both contain malicious data which comprises malware traffic. 115 features belong to each record that was produced by the publishers of the dataset through the raw characteristics of the network traffic. Here, both Gafgyt and Mirai traffic is combined to build the malicious data (968,750 records) since they are both generated from attack activity.

80% of the benign traffic data is given as input to the one class classifier for training the model. This means that 32, 316 records were used to train the model and (40,395 – 32,316) + (652,100 + 316,650) = 976,829 records were used to evaluate the performance of the model. This is also further given as input for the training of the autoencoder model. The rest 20% of the benign traffic data is sent to the testing phase of both the one class classifier and autoencoder model.

The Gafgyt and Mirai traffic data, which contains the malicious data is inputted along with the 20% of the benign traffic data to test the one class classifier and autoencoder models whether they help in sorting of the traffic into clean and malware traffic.

Pre-processing of data is done so as to improve the quality of the datasets involved. Since there are multiple datasets involved, functions are constructed to ensure that these multiple datasets are loaded to be entered into the required model.

Getting examples of cases with good traffic which displays benign behaviour is simpler and easier to get in comparison with malware traffic which displays malicious behaviour. This can be due to the fact that obtaining malware traffic can come at a cost or in some cases, impractical like the days with no attacks thus containing only good traffic. Other explanations regarding this could be on the basis of privacy, law and ethics. But, despite all this, it can be easily convinced that corrupted traffic data can be used first to build models like the two-class classifier. But it cannot be assured that all scenarios involving bad traffic data can be replicated. This issue can be addressed using the unsupervised one-class classifier methodology approached here.

So, in this project, the training model is created only using benign instances and this trained model is then implemented to detect any unknown/new cases of traffic using machine-learning and other statistical methods. If the data targeted shows considerable diversion according to predetermined calculations, it will be labelled as out-of-class. Thus, one-class classifiers that may belong to different families are examined here under the criterion of performance. The two one-class classifiers and their corresponding families shown here are One Class Support Vector Machine from the typical ML family and Autoencoder from the deep learning family. Here, we are considering that benign occurrences have a different kind of structure compared to that of corrupted occurrences. This structural difference is taken as the basis for creating the Autoencoder model.

One-class classifiers help detect instances of a particular class from all the other instances, through training sets containing only the instances of that class. The particular class considered here is a benign class. There are other kinds of one class classifiers where opposite instances are considered to sharpen the limit of classification.

Other purposes that a one class classifier can fulfil is in cases of binary and imbalanced classification datasets. In the scenarios where binary classification is to be done but the majority of the dataset overpowers the minority, the training set is modelled based on the majority category of data, before being jointly evaluated in the testing phase. For scenarios involving imbalanced classification datasets where the minority may be non-existent or not visible enough, supervised machine learning techniques will need to be implemented as one class classifiers are not designed for these purposes.

The support vector machine, or SVM, algorithm can be incorporated into one-class classification, even though its mostly meant for binary classification. In cases of imbalanced classification, the weighted and standard support vector machine can be applied on the dataset before one class classifiers are implemented. For one-class classification, the algorithm helps size up the density of the majority class and categorizes the extreme cases of the density function on either side as outliers. This variation of support vector machine is called a one-class support vector machine.

Autoencoder neural network is an unsupervised ML algorithm which incorporates backpropagation through keeping the required values same as the inputted values. This helps decrease the size of the input into reduced representation. The original data can be reassembled from the compressed data. The primary goal of autoencoders is to learn the representation of a dataset, mainly for the simplification of dimensionality, by training it to overlook noise data. The input is compressed and put into a latent-space representation and the autoencoders then reconstruct the output out of this.

The autoencoder comprises an encoder, code and decoder. The encoder helps compress the input into the latent space. The input image is encoded here as the compressed representation with a simplified dimensionality. Thus, this compressed version will be distorted in comparison to the original. The code part shows the input that was compressed which is loaded into the decoder. Finally, the decoder helps decode the compressed image back into the original version with the original dimension, resulting in a lossy reconstruction of the original from the latent space.

The features extracted through the above algorithms are then registered and stored in a database called the feature database. The machine learning classifiers included in the project are explained above. Thus, malware detection can be directly implemented resulting in allowance of the benign traffic to be sent and received through the device and removal of any transmission back and forth of malicious data found due to the presence of Gafgyt and Mirai datasets