Malware Classifier Using Machine Learning

*Note: Sub-titles are not captured in Xplore and should not be used

Katturu Manish, Kopalli Abhilash, Vattikuti Sai Sarath, Jyothula Sai Sandeep
Software Engineering Department, San Jose State University 1 Washington Square, San Jose, CA

manish.katturu@sjsu.edu
abhilash.kopalli@sjsu.edu
saisarath.vattikuti@sjsu.edu
saisandeep.jyothula@sjsu.edu

***Abstract*—This document gives an overview of the "Malware Classifier" project implemented for CMPE 272-Spring 2019. The objective of the project is to provide a platform for Security Administrators in every Organization to Classify a potential malware to a group which defines the threat level and remedies required to clean the malware. To achieve the objective, a classifier was developed which uses Machine learning algorithms to identify the malware file to its originated class.**

Keywords—component, formatting, style, styling, insert (*key words****)***

I.Introduction

In the recent years, the Malware market in the I.T industry has become a Multibillion Dollar Business, involving large sets of data. Multination Companies have been investing heavily on new antimalware products evading the traditional protection. This created a huge market for the antimalware vendors to develop potential counter mechanisms for finding the malware and deactivating them. There are prodigious amount of malwares existing in the world. Most of the malwares can be classified into nine categories i.e. Ramnit, Lollipop, Kelihos_ver3, Vundo, Simda, Tracur,Kelihos_ver1, Obfuscator.ACY, and Gatak. So, the identified malwares have various antimalware products built by the technological companies. But classification of thousands of malwares to the following classes has always been a problem in industry.

II. Problem Statement

Petabytes of data is generated daily by various anti-malware products by scanning 200M computers to identify potential malware. The multiplicity feature of a single malware file into a number of different files is one of the main reason for the high volume of data. Most of the malwares will manipulate their existence and look similar to the regular files. So, building effective techniques for classifying thousands of malwares to their respective families will help in preventing abundant loss to the companies by providing necessary solutions ..

A.Proposal

We are proposing to build a Web portal called Malware Classifier with a potential capability to solve the classification problem in the malware industry .This portal helps the System Administrators to identify the malware in a group so that the remedies can be directly applied to a whole group solving the issue in a short time. The Malware classifier also has the potential to save the malware files in the local database or in the Admins database which can be further used to train the machine learning classifier improving the accuracy of the algorithm. Nearly eleven thousand files with extension "bytes" that has an average size of 4.2 MB were used to train the algorithm. Each byte file consists of Hexadecimal code from which 256 features are extracted to train the machine learning model. Prediction of the Malware classifier will be based on the entropy value of the Tree generated from the Xgboost algorithm.

B. Scope

Malware classifier portal will provide a basic framework for classifying the malwares into 9 different classes as mentioned earlier. Portal is scoped specifically to help with classifying major malware families. Portal has a capability to extend in future for the following enhancements.

1. Using a wider dataset that covers all types of malware.
2. Keyword search of specific malware family features.
3. integrating our solution with dynamic detection techniques by profiling dynamic features for each category; dynamic features like system calls, network connections, resources' usage, and etc.

C. User Persona

Some of the real-world organizations suffered by malware are as follows:

1) Due to Misdiagnosing a malware type the Presbyterian Medical centers cookies are stolen which are highly sensitive for hijacking banks and their social media.

2) In 2016, Simda was reportedly found being used as a banking trojan. It was difficult to identify the malware and provide the solution. The Society for Worldwide Interbank Financial Telecommunication (SWIFT) used by many investment firms and banks is a transaction messaging network saw attacker targeting its clients, manipulating organizations in sending fraudulent money transfer requests.

3) In 2015, security researchers warn that the US healthcare organizations targeted by a trojan ,avoids detection after infection by being inactive for longer periods. Symantec, a cybersecurity software provider estimates that many organizations have been hit by the malware belonging to Gatak family, stealing the confidential data which was undetected by the anti-malware products.

III. Implementation

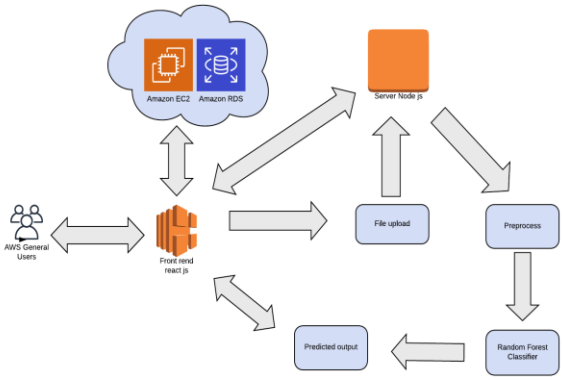

Fig. 1. Architecture Diagram

A.Front-End

For the front end design we relied heavily on Reactjs as it better suited for our needs,which is a single page application. With the help of google material UI icons and components, we have developed our home page, login and sign up pages for the customer. The pages render only in its state or props change in React. Since React uses virtual DOM, it is much faster and smooth from end user perspective.

A. Database

We have used a NoSQL database(MongoDB), because the entries gets saved as a collection and it supports JSON format unlike traditional relational databases.
MongoDB also supports javascript frameworks and easy to read and access. We used online cloud mongodb to host our MongoDB database on https://www.mongodb.com/cloud.

B. Back-End

Our obvious choice for implementing the server side scripting is NodeJS, as it is a light weight process, non-blocking and due to its asynchronous nature. And moreover it integrates well with ReactJS and it is easy to implement it because Node and React use Javascript methodologies (request and response) for sending and receiving data from frontend to backend, vice-versa.. It is rich in libraries, thanks to Node Package Manager and due to its scalability. We used express, passport, jwt-decode and express-fileupload, to mention a few.
We hosted our backend on static IP, where we bought a domain for hosting the code.

D.Classifier Model

Malware Classifier is a machine learning model that is based on the concept called supervised Learning .We have identified the problem as a multiclass classification problem The model that we trained using classification algorithms ,classifies a given malware file to one of the 9 classes from the trained data .

In our proposed work Building the model involved several steps:Firstly, the dataset which are byte files are converted to text files for extracting the data.From each textfile features are extratced for training machine learning models.The new dataset which is in the form of (.csv) has undergone Min-Max Normalization to bring the whole data to a scale of 0-1.Then,the data is split for 80 percent training data and 20 percent testing data .Out of the 80 percent training data 16 percent is further trimmed for crossvalidation.So,now for the given model the testing data is passed to predict the accuracy.

We have used 4 machine learning models out of which XGboost got the best accuracy and the least logloss.The given table shows the accuracies and log loss for each of the four algorithms.

The log loss function defines the throughput of the model. The lower the value the better the model

$$logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} yij \log(p_{ij})$$

where N is the number of files in the test set, M is the number of labels, log is the natural logarithm, yij is 1 if observation i is in class j and 0 otherwise, and pij is the predicted probability that observation i belongs to class j.

S.NO	Machine Learning Models		
	Algorithms	Accuracy (%)	Log loss
1	Random Forest Algorithm	96.2	0.0821
2	K Nearest neighbhours	95.23	0.2020
3	Logistic Regression	89.16	0.4957
4	XG Boost Algorithm	97.7	0.0792

Fig. 2. Model outputs

PROCESS FLOW

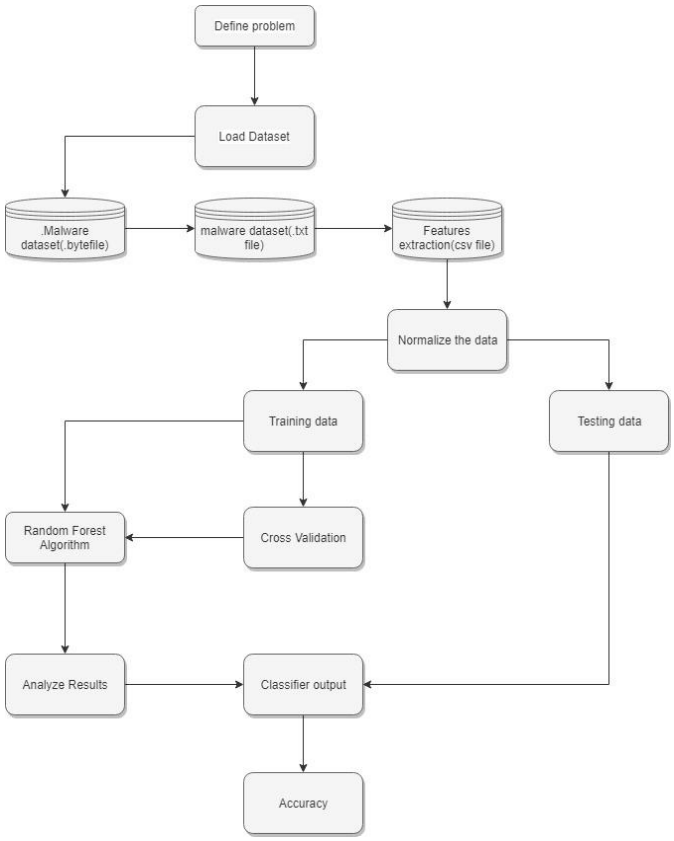

Fig. 3. Work Flow of the model

IV.FUTURE ENHANCEMENT

Provision for the user who intends to check, in which class a particular malware class a file falls in by uploading any type of file, it only determines for a byte file as of now. Further, we also want to design the model to train by itself(continuous learning) from all the inputs provided by the user using Apache Kafka. We would like to upload multiple files at once rather than one file at a time. In the future, we would like to embed our software along with other anti-malware detecting softwares where in our software would predict to which class that malware belongs to.

V. Conclusion

We have identified that there is lot of scope for machine learning in the malware industry. By using the available data in the malware industry, the antimalware products can be further improved .So, the technologies that are built to erode the malware will be used to the outputs obtained through our classifier.

Inspired from prof. Ranjan idea's in developing a project where there is real scope and need for development, We tried to achieve a web portal which classifies a malware file to a parent class. This web portal bolsters the existing technologies. We have achieved 97.7percent accuracy in our model to classify the files to its Root class. With the classifier we have built the crucial time between the identification and application of the anti-malware products will be saved and the versatile damage that will be done during the scope will be abated. So, when we implement this classifier along with the anti-malware products, it fetches better scalability and throughput to the product which in return captures the Business market and the user's credibility for using the products.

VI. Contributions

Vattikuti Sai Sarath-Student in MS Software Engineering, SJSU. Developed the xgBoost algorithm with hyper parameter tuning and also developed KNN algorithm. Apart from that I have preprocessed the data for extracting the features from the byte files. I have identified outliers in the data and eradicated them using normalization techniques. I have used 2 normalization techniques: z-score normalization, min-max normalization out of which min-max normalization is best suited for our problem. KNN obtained its best accuracy with K=3 which showed a log loss of 0.2020, whereas xgboost showed a log loss of 0.0792 with 500 estimators. On top of these, I have also developed the code for File upload functionality which classifies the input into a Class.

Jyothula Sai Sandeep-Student in MS software engineering. Developed Random forest algorithm with hyper parameter tuning for our project. Also performed analysis in types of malwares for understanding the features to use for training the model. Random forest algorithm performed best with 2000 trees obtaining the log loss 0. 0821.Apart from that I played role in deciding the metrics for our model evaluation and also developed code for confusion matrix function using heatmaps. I have also designed the models for deployment pickled it using the pickle and cpickle.and stored them in the cloud.

Manish Katturu- Student in MS software engineering. Developed and implemented logistic regression. Which has log loss of 0.495 I have used advanced optimization algorithms like minimum of unconstrained multivariable function. Also implemented and handled the database connection and hosted the Mongodb database in AWS cloud for storing the user login details and upload history of user.

Abhilash Kopalli- Student in MS software engineering. Developed frontend views for homepage, signup and login pages for the user using ReactJS. Using routes in react for navigating through pages. And developed the back-end functionality for the user using NodeJS and handling session using express and using the jwt-decode to decode JWT's tokens. And also wrote the file upload functionality to upload a file by the user and execute a python script in a backend to determine which class a particular malware belongs to.

VII.Project Repository

- Malware Classifier project code-base, available at https://github.com/SJSU272Spring2019/Project-Group-21 .

VIII.References

[1] Apache Spark™ - Unified Analytics Engine for Big Data,

[2] MLnick. “MLnick/Elasticsearch-Vector-Scoring.”GitHub, github.com/MLnick/elasticsearch-vector-scoring.

[3] “Project Jupyter.” Project Jupyter, jupyter.org/.

[4] https://www.kaggle.com/c/malware-classification/overview/evaluation

[5] https://digitalsecurityworld.com/9-types-malware-may-put-data-risk/

[6] https://www.beckershospitalreview.com/healthcare-information-technology/9-types-of-malware-that-should-be-on-a-hospital-s-radar.html