Malware Detection Using Machine Learning

Kapil Sharma RA1711003010542
Computer Science of Eng.
SRM Institute of Science and Tech.

Anand Kumar RA1711003010638
Computer Science of Eng.
SRM Institute of Science and Tech.

R. Radha – Asst. Professor
Computer Science of Eng.
SRM Institute of Science and Tech.

Abstract- In this age of Internet, all import data are shared between computer over network. The major concern of sharing data is security and privacy which have threat from malicious software called as Malware. In order to provide security, first step is to detect it before making adverse effect on our system. There are many types of malwares based on the features like the way they effect and spread in our system. In order to detect it very first step should be to classify it on its features.

In this paper we will classify 9 types of malware by using Data Set released by Microsoft in 2015 for malware detection. This dataset contains more than 20,000 malware samples. All its features are derived from command line inputs which makes it more unique. After surveying many research papers, we find that ensembled based learning with CNN gives best result when it is used with machine learning models like Gradient Boosting and Extreme Gradient Boosting. Also, we found that final F-Measure will give approximately 0.97 scored when checked with normal malware files.

I. INTRODUCTION

As the number of malware attacks have increased since last two decades it has become extremely important to detect the malware quickly and accurately. And by using current improvements in the field of AI and ML this process of malware detection can be made much more efficient and faster.

Main focus of the survey is to find most effective and efficient way of classifying malware with given dataset.
Using advance concepts of Neural Network with added machine learning features help us to classify as well as detect malwares.

Index Terms
- Malware Detection, Machine Learning

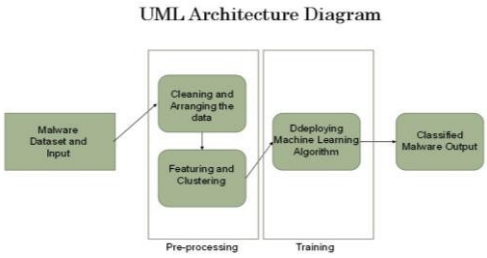

Fig. 1. UML Architecture Diagram of Malware Detection Using Machine Learning.

II. STATE OF THE ART (INFERENCE OF LITERATURE SURVEY)

1. The major goal is to keep malware away from Google play Store itself. The main focus of this paper is to Develop and effective malware detection system by selecting best features of the malware's. It also tries to judge that what is the usefulness of given feature in detection of malware

2.In recent time many mobile based, deep-learning based and IoT based approaches emerges in recent time. It also depicts the major challenges i.e. signature based detection is not possible and in behavioural based detection many features are required for classifying malware. The conclusion they found that no machine can detect new generation malwares with full accuracy.

3..The older malware detection techniques takes long time and they are less efficient. At same they proposed new model where features are derived using Convolutional Neural Network and latter it is classified using SVM. On evaluation the accuracy gained is 0.997. This model is much effective and fast than older ML algorithms.

4.This dataset is becoming a standard dataset with more than 50 papers citing it. We will take all these references as far as possible and compare others main inferences in respect with the dataset. The comparison helps to let us know the valuable inferences and the potential of research directions leading to more research.

5.More investigation on this data will be required to give the idea of effectiveness of neural networks as classifiers for structured dataset when tallied with other models. Real world testing is a good to research on because as shown that there are still chance of betterment and clear loop holes are present that not been reached or searched so far. Thus, it will require further research for real time implementations.

6.Since there aren't get these problems in these papers we have found that it is very important to found the common features in the data which are mutually exclusive this will help in classification of malware correctly .

7.Malware Analysis can be done by using deep learning algorithms as well. Using these algorithms are becoming quite common nowadays. Thus, it is obvious to take static and dynamic analysis properties with deep learning neural networks for Microsoft Malware Classification.

8.CNN-SVM, GRU-SVM, and MLP-SVM and many other models are used for classification. There are many papers that has proved that the GRU-SVM comes out more accurately within all other Deep Learning Models It come out with a predictive accuracy of 84.92.

9.N-grams also become famous way which is commonly used to get back all information and machine learning applications from decades. High N-grams are much more beneficial in extracting properties which are useful by malware analysis, and by general industries tools like YARA it is possible to make general purposes signatures.

10. The models presented in this paper ,using emulation sequential data is derived.

11.Two generally used models of deep learning ie.CNN and LSTM together is used for giving impact to end to end learning of models

12.The Extended Burrows Wheeler Transform (EBWT) build explicitly around the Burrows Wheeler Transform which is based on comparison based on distance metric. algorithm for use in bioinformatics.

13.All authors aim to keep the reference table updated and encouraged us to cite these papers while using the dataset.

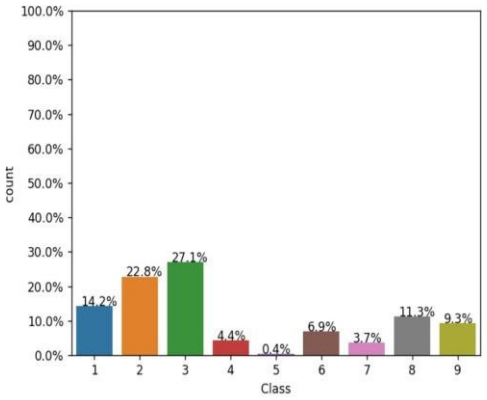

Fig. 2. EDA On class distribution of complete dataset

Fig. 2 shows distribution of classes in complete dataset

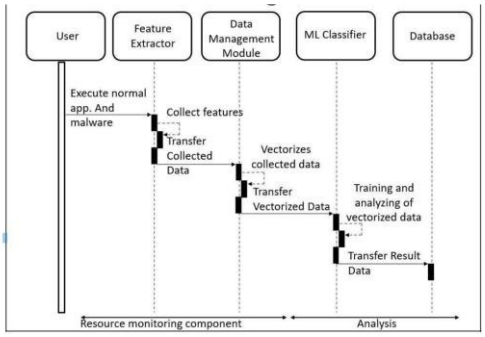

Fig. 3. UML Sequence Diagram

III. List of Abbreviations

1. UML - Unified Modeling Language
2. SVM - Support Vector Machines
3. CNN - Convolutional Neural Network
4. LSTM - Long Shot-Term Memory
5. EBWT - Extended Burrows Wheeler Transform
6. DLMD - Deep Learning based Malware Detection

IV. PROPOSED WORK

Fig. 1 Shows the UML architecture diagram of the complete process and

A. Data sources and Data Description

Microsoft company has its own set of antivirus software which they deploy of millions of computers worldwide, due to which they generate millions of data points and peta bytes of data. We will be using that data in out research.

The data will have 2 types of files in it namely:byte files and assembly language files. Both of them are equally useful in determining which type of malware the data point belongs to.

The complete size of data will be 200GB out of which around 50 GB is hexadecimal byte code data and the rest is assembly language code data.

Classes of Malware:

1. *Ramnit*
2. *Gatak*
3. *Tracur*
4. *Vundo*
5. *Obfuscator.ACY*
6. *Kelihos_ver1*
7. *Simda*
8. *Kelihos_ver3*
9. *Lollipop*

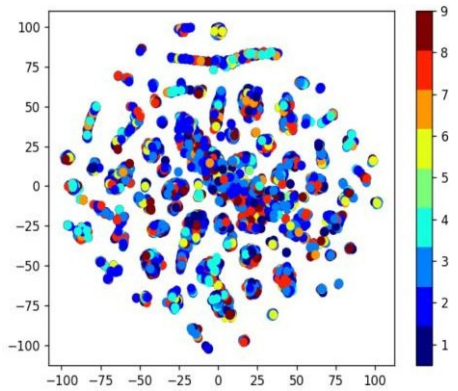

Fig. 4. multivariate analysis on features before feature engineering

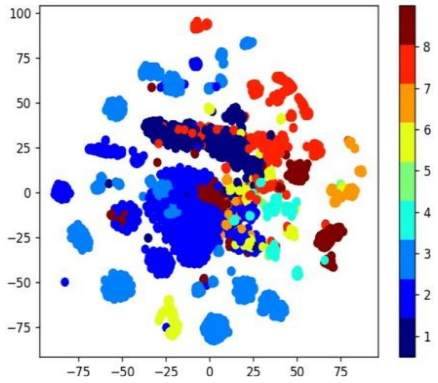

Fig. 5. multivariate analysis on features after feature engineering

A. Feature Engineering and Exploratory Data Analysis:

First, we have done feature engineering on the data **[Fig.4]**set to get unique mutually exclusive features out of the data. After doing feature engineering we have plotted the multivariate analysis of final features **[Fig.5]**

After doing Feature Engineering we can see that classes are now more separable from each other which will lead to higher accuracy by classification algorithm;

Fig.3 shows the UML sequence diagram of the complete process

Conclusion on EDA of data after Feature Engineering
We have taken only 52 features from asm files (after reading through many blogs and research papers)
The univariate analysis was done only on important features.
Take-aways
1. Class-3 malware values are differentiated because of the frequency of keywords being fewer
2. Each feature has its unique importance in separating the Class labels.

B. Machine Learning Algorithms:

As inference above, deep neural networks is not the best choice for malware classification as it becomes difficult to inference the properties of malware after classification due to black box nature of deep neural nets and it also makes model complex and heavy.

Therefore, we will use SVM, KNN, Random Forest, XG-Boost Classifier **[Fig.6]** for our classification models. After doing

hyperparameter tuning on XGBoost Classifier it gives best accuracy among all classification models. The results are shown in **[Fig .7]**

```
x_cfl=XGBClassifier()

prams={
    'learning_rate':[0.01,0.03,0.05,0.1,0.15,0.2],
     'n_estimators':[100,200,500,1000,2000],
     'max_depth':[3,5,10],
    'colsample_bytree':[0.1,0.3,0.5,1],
    'subsample':[0.1,0.3,0.5,1]
}
random_cfl=RandomizedSearchCV(x_cfl,param_distributions=prams,verbose=10,n_jobs=-1,)
random_cfl.fit(X_train_merge, y_train_merge)
```

```
print (random_cfl.best_params_)
```

```
{'subsample': 1, 'n_estimators': 1000, 'max_depth': 10, 'learning_rate': 0.15, 'colsample_bytree': 0.3}
```

Fig. 6. Code for training data with XGBoost Classifier on final features with best hyper parameters using Random search

```
x_cfl=XGBClassifier(n_estimators=1000,max_depth=10,learning_rate=0.15,colsample_bytree=0.3,subsample=1,nthread=-1)
x_cfl.fit(X_train_merge,y_train_merge,verbose=True)
sig_clf = CalibratedClassifierCV(x_cfl, method="sigmoid")
sig_clf.fit(X_train_merge, y_train_merge)

predict_y = sig_clf.predict_proba(X_train_merge)
print ('For values of best alpha = ', alpha[best_alpha], "The train log loss is:",log_loss(y_train_merge, predict_y))
predict_y = sig_clf.predict_proba(X_cv_merge)
print('For values of best alpha = ', alpha[best_alpha], "The cross validation log loss is:",log_loss(y_cv_merge, predict_y))
predict_y = sig_clf.predict_proba(X_test_merge)
print('For values of best alpha = ', alpha[best_alpha], "The test log loss is:",log_loss(y_test_merge, predict_y))
plot_confusion_matrix(y_test_asm,sig_clf.predict(X_test_merge))
```

```
For values of best alpha =  3000 The train log loss is: 0.0121922832297
For values of best alpha =  3000 The cross validation log loss is: 0.0344955487471
For values of best alpha =  3000 The test log loss is: 0.0317041132442
```

Fig. 7. Code for results after training with XGBoost Classifier

V. RESULT DISCUSSION

As shown in Fig.8 and Fig.9, By using XG Boost Classifier and doing hyper parameter search for best hyper parameters on our final data set which we got by doing feature engineering on the given data. We reduced the log loss to minimum possible value **[Fig.8]** and with the maximum possible accuracy as shown in the confusion matrix [**Fig.9]**

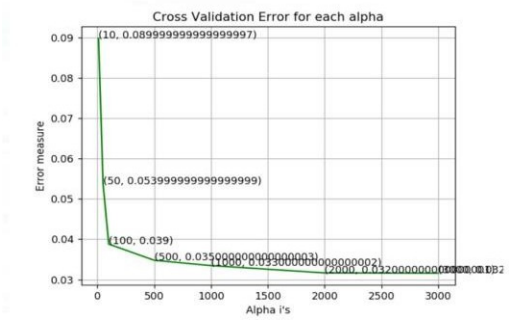

Fig. 8. Log Loss Reduction Graph

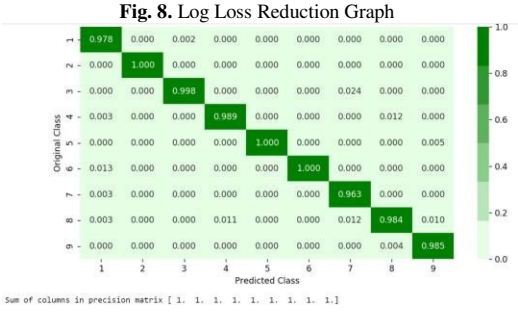

Fig. 9. Confusion Matrix of results after training final data on XGBoost Classifier

VI. CONCLUSIONS

We have concluded that Tree based classifiers such XGBoost and Random Forest will classify the given malware file into its particular malware class in the least possible time with the most possible accuracy.

The reason behind this is the lesser number of features for the amount of data given due to which we have less height and branches of tree which reduces the time and increases the accuracy respectively.

Some more research on the data can help in understanding how deep learning based neural netwroks can be used in classification if the intent is not to get inference from the classification as neural network models tend to black box in nature.

VII. Future Scope

1.Bi-grams and N-gram features can be added on hexadecimal code so that error value can be reduced even more.

1.Image Features can be used to improve log - loss further as used in the reference below https://github.com/dchad/malware-detection

VIII. BIBLIOGRAPHY

[1] Saint Yao ,2019, IEEE, In Integration of Static and Dynamic Analysis for Malware Family Classification with Composite Neural Network, we learned Deep learning has been used in the research of malware analysis. We combine static and dynamic analysis features with deep neural networks for Windows malware classification.

[2] Mansour Ahmadi, 2018, ELSEVIER, Novel Feature Extraction, Selection and Fusion for Effective Malware Family Classification, we learned Modern malware is designed with mutation characteristics, namely polymorphism and metamorphism, which causes an enormous growth in the number of variants of malware samples.

[3] Swati Agarwal ,2019, Springer, In Malware Classification using Deep Learning based Feature Extraction and Wrapper based Feature Selection Technique we learned A deep learning based malware detection (DLMD) technique based on static methods for classifying different malware families. features are extracted from byte files using two different types of Deep Convolutional Neural Networks (CNN).

[4] Joshua Saxe 2017, IEEE Transactions on Neural Networks and Learning Systems, Deep neural network-based malware detection using two-dimensional binary program features.

[5] Ivan Firdausi, 2017, IEEE, Intelligent Systems, Dynamic data fusion using multi- input models for malware classification.

detection using machine learning Automatic malware classification and new malware detection using machine learning.

[6] Piyush Aniruddha Puranik 2019, IEEE, Intelligent Systems. Towards Building an Intelligent Anti-Malware System:
A Deep Learning Approach using Support Vector Machine (SVM) for Malware Classification

References

Reference Papers:
Paper 1: https://ieeexplore.ieee.org/document/8268747
Paper 2: https://www.researchgate.net
Paper 3: https://ieeexplore.ieee.org/document/8268754
Other Papers:
https://paperswithcode.com/task/malwareclassification

Reference for Dataset:
https://www.kaggle.com/c/malware-classification
https://www.microsoft.com/security/blog