OPEN ACCESS

Kashf Journal of Multidisciplinary Research
Vol: 02 - Issue 1 (2025)
P-ISSN: 3007-1992 E-ISSN: 3007-200X
https://kjmr.com.pk

MALWARE ANALYSIS AND DETECTION FOR MICROSOFT TECHNOLOGIES

Muhammad Ahmad Shahid*
Department of Computer Science, Government College University, Lahore, Pakistan

Muhammad Safyan
Department of Computer Science, Government College University, Lahore, Pakistan

Abdullah Mustafa
Department of Computer Science, Pakistan Embassy College, Beijing, China

Corresponding author: ahmad.shahid.129@gmail.com

Article Info

Abstract

Malware detection is always a hot issue and a priority task in cyber-crimes. Despite a lot of work in the past malware detection in Microsoft Word is being a major challenge for researchers and other practitioners. This research examines the malware and detects the malicious files with the help of structural path features and lexical based features of extracted URL from unzipped XML files of Microsoft Word. This research carried out three experiments and finally reached to a goal with 0.97% accuracy with a highest true positive rate of 0.98% and lowest false positive rate of 0.012%. It showed a somehow reduced TPR rate in detecting benign files but can be increase in future while doing more precise work upon malicious URL used in documents.

Introduction

In this modern world and growing needs of Internet Of Things(IOT) malware detection has become challenge both for anti-malware company and a researcher. Cybercriminals have financial support and exploit the users by deliberately damaging them, stealing their personal information. And most importantly that the malicious documents are using immensely for injecting the malware into the system and this is all made possible by the negligence and trust of user upon the software and poor detection of antivirus malwares.

In 2006, Microsoft vendors introduce its new format for its 2017 Microsoft office product that is OOXML format. This Open XML format includes many new specifications for example XML files enclosed in a ZIP format, compression of files, less memory space etc. [16]. Now a day XML is playing an important role in the exchange of files, as the major application now support XML format. Many XML supported applications are now available over the internet and many vendors are also using it [2]. With this new format most of the users and vendors are using this OOXML format but with its growing demand its faces many threats and vulnerabilities due to transfer of files. Because many criminals and attackers are sitting beside to inject the OOXML format of DOCX with malicious code or hide malicious data or inject them with malicious payloads. With the tremendous usage of Microsoft Word there is an urge to have more research and work against its threats and attackers. Therefore, malware detection in systems and applications is one of the major and first priority tasks of cyber-security. So we need a classifier that will tend to improve the detection rate by using best algorithm of machine learning as there is least work done for detection of malware in Microsoft Office and almost no research is done in past for detecting the malicious links inside the documents that looks benign but perform many malicious activities without user knowledge.

In Fig.1, statistical bars showing the new malware trend growth of last 12 months. Therefore, machine learning is a method that has a strong capability to malware detection and therefore competing in the World for detection of known and unknown malwares.

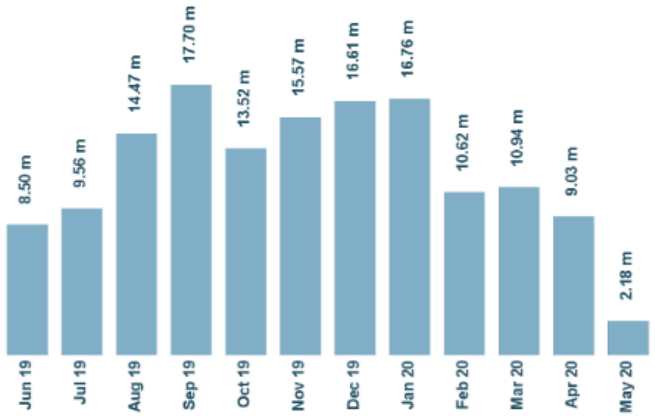

Fig. 1. Statistical representation of NEW Malware In 12 months [1]

I. Related Work

Since 2019, it is reported that about 60% of email attachments and 20% of other files that infect the system are MS Word files, MS excel files, and PDF. These document based malwares not only used to spread malicious activities but also greatly used against government [7].

Alazab in his paper discussed the advanced machine learning algorithms and their efficiency in detection of malwares. As malware threat is increasing day by day and becoming the most important topic of this internet World. They evaluated the machine learning algorithms along with deep learning algorithm that apply both static and dynamic techniques along with image processing in the detection of malware and develop a strong framework called ScaleMalNet[8]. This paper doesn't discuss about its behavior in adversarial environment. Lin performed the extraction and selection of features for efficient malware classification through machine learning methods. Lin and Wang discussed that the most widely used static based technique i.e., signature-based technique for malware classification does not encounter the proper classification of malware and unable to detect the unknown malwares. In this case dynamic malware classification technique can fulfill this need [4]. In this paper Lin and Wang put forward a behavior-based technique and propose an algorithm that works in five steps. Behavior of malware was obtained from the sand box environment. First step of algorithm include extraction of features through n-gram feature selection technique, in the second step formation of support vector machine for classification, in the third step features are selected for efficiency of algorithm, in the fourth step high dimensionality features are converted into low dimensionality feature scale and in the last step finally a model is built for the malware classification. Furthermore they propose their method for online training simulation that reduce the time cost and also increase the efficiency [4].

MS Office files like .doc, .docx, .docm have more threats and vulnerabilities of malware and to get infected through malicious activities. The basic reason for having more threats to these files is the use of malicious macros that are embedded within the MS Office files [5]. Cohen elaborates and performed experiments to detect the malware in MS Word documents. Author uses the structure featured extraction method (SFEM) along with machine learning as according to author in past no work has been done on analyzing and detecting the malwares from OOXML formats documents that are .docx, .docm, xlxs etc. [6]. Cohen and Aviad used SFEM along with machine learning classifiers to detect the unknown malwares. They use a large dataset of MS Word documents and outperformed four different experiments upon them. All the experiments showed that SFEM along with machine learning techniques give better, fast and speedy results [6]. But despite of its accuracy it doesn't detect the hyperlinks that refer a document towards a malicious website and thus leave those malicious files as benign. And the second thing to be noted that changing parameters of features extracted or new features about which the classifiers is not trained upon affect the file structure and remain undetected by the classifier. Thus the model generation needs to be regularly updated upon which this future work can be done. And in the future this type of work and experiment can be performed upon the Meta features.

Web weaknesses are on the ascent with the utilization of cell phones and computers for both individual and proficient use. With the excessive use of web, vulnerabilities are increased to much extent that unauthorized users are using clickable links inside the documents that install the payloads or perform any malicious activity that take the user onto the other site without the permission of user. Chong and Liu uses the lexical features of URL to detect these malicious URL [13]. This paper centers around an AI arrangement that recognizes noxious URLs utilizing a blend of URL lexical highlights, JavaScript source highlights, and payload size. Chong uses the machine learning approach with SVM to detect the malicious URL and achieves 0.81 accuracy rate through polynomial kernel [13]. Malicious activities on web sites have been grown to an alarming level on internet. And in return these malicious URL activities are also performing inside a document. Michael Darling says that there must be a need of intelligent systems that can detect the malicious URLs. In this regard analyzing a URL for malignant activities can give better results in detecting obfuscated, malicious and phishing uniform recourse locators [9]. Michael uses the lexical features alone to build a classifier that provides the accuracy rate of 99.1% which outperforms all other with a low FPR of 0.4%. URL shortening administrations are getting progressively famous both with aggressors and the overall population. At the point when a client taps on an abbreviated URL, it

diverts to the full length URL of the page. But this [9] paper only tackles the URL with full length URL feature. Frank in his paper [12] detects the malicious URLs using binary classifier in machine learning and compared the results of these different classifiers

Microsoft office files like .doxc, .xlsx, .pptx are immensely using by the attackers for spreading malicious content. Microsoft Word which is using greatly by the attackers for spreading malicious content, most commonly through emails, phishing attacks, malicious links, enabling macros, embedding OLE objects etc. Thus it concluded that there must be a classifier that can classify the malicious documents based on structural features of document and lexical features extracted from URL used in document. URL analysis must be done statically so that we don't have to visit the malicious websites.

II. THE PROPOSED METHOD

Proposed methodology includes all the necessary steps and procedures that are needed to achieve our goal in detection of malware with greater accuracy. The Fig. 2 is the detailed view of our proposed architecture.

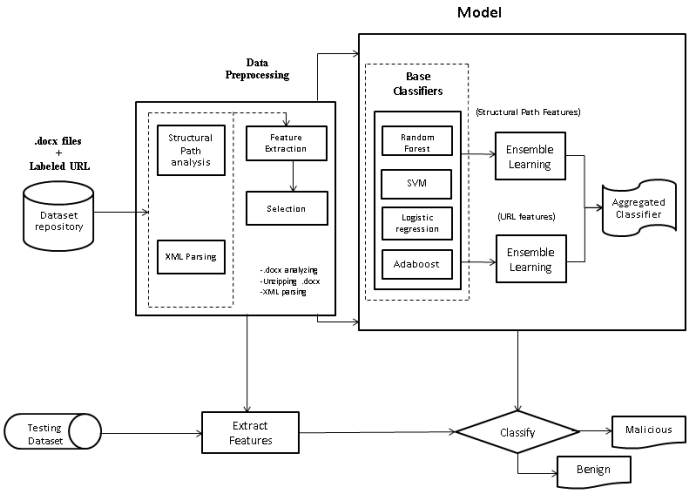

Fig. 2. Proposed Architecture

A. Data Repository

Data is the most valuable asset of any organization and searching of accurate dataset for your required experiments is much difficult task. Dataset for our proposed methodology was collected from various sources. These sources include the Virus Total[3], Virus Share[10] and Conatgio[11]. Major benign files were collected from Virus Share and contagio and malignant files were collected from Virus Total site. We have collected the major malicious data files that have been exploited and vulnerable since 2017 so that we can achieve more meaningful results and can have more true positive rate. We have collected total 4525 benign files and 3028 malicious files altogether from all resources. All the files are confirmed under the label of malicious and benign through the reports from Virus total. These all files (both malicious and benign) are stored with MD5 hash function which described that all the files are created originally and doesn't include any modification since its publication on site. The reports that we have generated from virus total are in JSON format and through python library we have checked these json files and confirmed that all files that we have collected are in OOXML format and all files have their MD5 value. We analyzed that most of the malicious files are malignant due to exploit vulnerabilities commonly are (CVE-2017-8759), (CVE-20170199), (CVE-2017-11882) etc. These vulnerabilities depict that the most of malicious files in our collection ranges from the recent vulnerabilities since 2017. We have also analyzed that majority of our collected malicious files belong to the Trojan family. We have collected two types of dataset for our proposed methodology. One dataset consist of benign and malicious documents of .docx

files and other dataset of malicious and benign Uniform Resource Locator (URL) so that we can train the classifier upon the malicious and benign URL and then can used that model to extract and predict the malicious URL from documents that was externally used. We have collected both the malicious and benign URL from an online source. We have collected total 857 URL, among which 329 are malicious and 528 and benign URL. So two different dataset will be used to train the classifier.

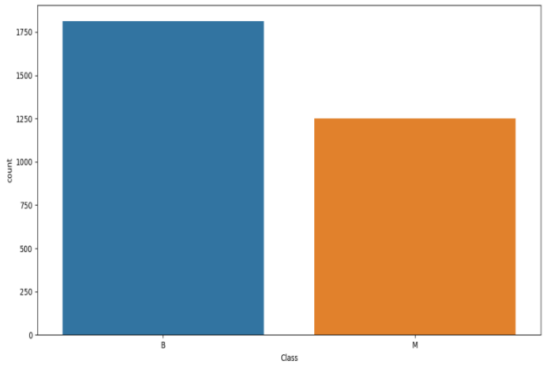

Fig. 3. Graphical representation of benign and malicious files

B. Data Preprocessing

This section includes a detailed description of the entire data set, which was gathered from various online sources in raw form before being prepared for use in an experiment. Both benign and malicious files are collected from different resources like Contagio, Virus total and total 2965 unique paths as features are extracted both from both benign and malicious files to perform the experiment based upon proposed model. Dataset that was collected is as follows:

Table 1 Dataset distribution and extracted features

	Malicious files	**Benign Files**	**Total files**	**Extracted Features**
Data set(.docx)	1251	1813	3064	2965
Data set(URL	329	528	857	22

According to the table 1, the total number of malicious and benign files is 3064.The paths of XML files from the OOXML structure of Microsoft Word documents with the.docx extension were used as the strategy for feature extraction. The OOXML format of Microsoft word contains a zipped folder of different files and subfolders and thus forms a root tree of files and folder as shown in Fig.4.

We have used the python environment and its library to unzip all the files in our collection and parsed its XML files for analyzing the document thoroughly. As all the XML files have unique paths so we have extracted all unique paths by ignoring the tags inside the XML files. These tags can also depict the useful information but extracting all tags and their values can take a lot of time and can make extraction process even more complex and slow.

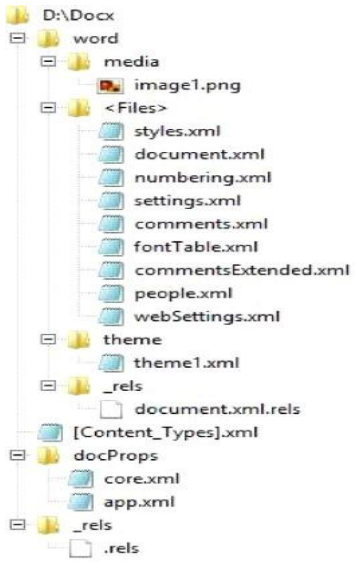

Fig. 4. Root Tree of an unzipped document

```
In [1]: runfile('C:/Users/Guddu/Documents/pars0.py', wdir='C:/Users/Guddu/Documents')
['[Content_Types].xml', '_rels/.rels', 'word/_rels/document.xml.rels', 'word/
document.xml', 'word/theme/theme1.xml', 'word/settings.xml', 'word/styles.txt', 'word/
stylesWithEffects.xml', 'docProps/app.xml', 'docProps/core.xml', 'word/fontTable.xml',
'word/webSettings.xml', 'word/numbering.xml']
```

Fig. 5. Extracted path list of a single .docx document

The Fig. 5 is showing all the list of extracted paths from a single file. This is a list of total 13 unique paths extracted from a file. The number of unzipped XML files can vary depending upon the document, its embedded objects and its content. We have collected total 3064 both benign and malicious OOXML documents with .docx extension. So total features extracted altogether from benign and malicious files are 2964.

Without accessing the content of websites, our mechanism only analyzes the Uniform Resource Locator (URL) itself. It thus removes run-time lag and the potential to expose users to bugs depending on the browser. An unzipped Microsoft word file contains many links inside different files. Every file has some links that point to other files that store the id and are linked to that file. Reels file is an unzipped docx file that shows the relations among the files and their corresponding ids. We discovered that if we change the link of relationship file manually, then docx file become corrupted and unable to open but while opening Microsoft Word recovers the content and it automatically correct the relationships links so altering or changing these links will not give any benefit to an attacker. After complete analysis we found that if the file used an external link to a website or email or the image that contains a hyperlink there information is stored in document.xml.rels file of an unzipped folder. When a document.xml.rels file of docx file, having an external links and hyperlinked images, was parsed it was found that their corresponding relationship ids, types, target links and mode are present there and an attacker can deliberately use them to perform a malicious activity. These links whose Target Mode is external can be used to check for a malicious activity so we use them and extract those links whose target mode is external and extract the lexical features from them. Total 22 unique features are extracted both from the benign and malicious URL. Three types of features from the URL are extracted:

- lexical features

- site popularity features
- Host based features

C. Feature Selection

Data without cleaning and refinement can create noise and may produce over-fit or under-fit results. There are lots of features that we have extracted and need to be redefined in order to avoid noise and over-fit results. Our dataset of .docx files consisted of three thousand and sixty four (3064) files among which 1251 were malicious and 1813 were benign with shape (3064, 2964).

Our methodology employed the Boruta algorithm, which is based on a random forest algorithm and selects the features that have larger gain and more importance than the other features of the dataset. It creates another data frame of X features and shuffles them all which are called as shadow features of X. Then it compare the feature importance of original features X with threshold value and threshold is the maximum feature gain obtain from a shadow features. And a "hit" occurs if feature importance is greater than threshold. Store feature importance as:

feat imp X = forest. Feature importances [: Len(X. Columns)] *(3.1)*

feat imp shadow = forest. Feature importances [Len(X. Columns) :] *(3.2)*

Compute hits:

hits = feat imp X > feat imp shadow.max() *(3.3)*

Before feature selection we had total 2964 features. These features must be reduced to minimize the complexity, training time and to avoid over fitting of data. We applied Boruta feature selection upon the feature dataset extracted from Microsoft Word files. Thus after applying Boruta algorithm we obtain the reduce feature set of 84 features, thus the new dataset is with shape (3064, 84).

III. MODEL

We build an ensembles learning based model which uses multiple machine learning algorithm as base classifiers and provide the combine results from these algorithms. We used four machine learning algorithms these are SVM, random forest, logistic regression and adaboost. Through maximum voting technique we trained our ensemble model with training set (X, Y) where input as X and output classes as Y:

$input = x_1 \ldots, x_n, output = y_1 \ldots, y_n$ *(3.4)*

Our input "X" in training set contains the features both structure paths and tags features extracted from Open XML format of Microsoft Word files. These features are trained upon two different datasets prior upon the Microsoft Word files with .docx extension and latter one upon the dataset of benign and malicious URL's. Whereas, "Y" contains the class label 0 and 1, as "0" represents the benign class and "1" represents the malicious class. Thus we trained and build two different models for structure paths features and XML tags features. For classification questions, multiple models are used in this methodology to make predictions for each data point. We used different types of machine learning algorithms as level-0 that is base model for predictions and ensemble learner as level-1 called meta-model. We pass our input array X unto the set of base models, each fed with input array X as;

y = f(x, a) *(3.5)*

where “a” is the classification parameter. Every model’s prediction is regarded as a vote. As the final forecast, the predictions we have from the most of the models are used as the last prediction from an ensemble model. We applied both soft voting and hard voting in order to achieve maximum results and perform comparisons among them while weights are kept uniform.

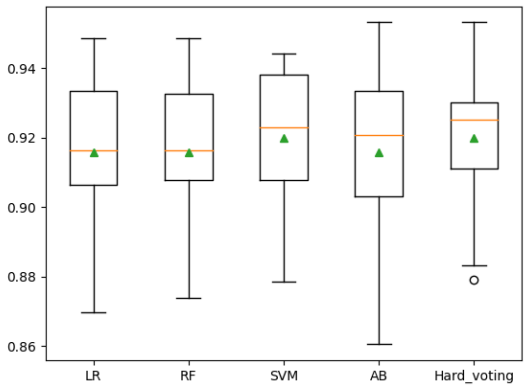

Fig. 6. Hard voting ensemble learning for Structural path features

The Fig. 6 and Fig. 7 are the comparisons among the soft voting and hard voting ensemble learning among the heterogeneous base learner’s. Both shows that ensemble voting and SVM depicts more accurate results than other models in case of structural path features. Thus we choose to use the hard voting ensemble technique to build model as it surplus the results of soft voting because in our case there are binary labels of target value and also our dataset contains sparse data. While Random Forest works best when features are in mix form that is when both categorical and numerical but our dataset is in binary representation. Logistic regression also failed to reach the greater accuracy due to large feature vector space. We used supervised learning and reallocate the weights that are comparatively higher and minimize the variance as well as bias. Adaboost is another machine learning technique that is used as an ensemble technique. This algorithm boosts up the slow learners and compute better results than other ones. We analyze that when we used adaboost it further improves the results and boost up the prediction results when used in ensemble learning to train our feature vector space.

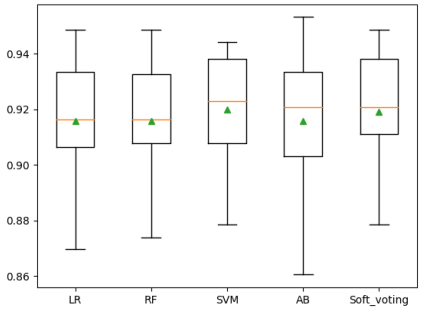

Fig. 7. Soft voting ensemble learning for Structural path features

In case of URL features, hard voting and soft voting depicts same accuracy scores along with Random forest as in this case Random Forest surplus the results over SVM because our training dataset for URL is in mix form that is both categorical and numerical feature vector space while SVM works well with homogenous features and also with higher dimensionality data. So in both cases, we used the ensemble learning with hard voting to increase accuracy rate and to reduce both bias and variance to achieve our

goal of detection and classification. We performed three experiments using the structural path features and Lexical features of URL to reach our goal with our proposed methodology. First experiment was done with path features, second experiment was done with URL features and third and final experiment was performed using these two trained classifier upon ensemble learning and applied logical reasoning upon results.

IV. RESULTS & EVALUATION

We build an ensembles learning based model which uses multiple machine learning algorithm and provide the combine results from these algorithms. Three experiments were performed to reach our goal. First experiment was done with path features, second experiment was done with URL features and third experiment was performed using these two trained classifier. We have conducted the following three experiments in order to find the answers to our proposed research questions that either

1. Our methodology provides best results to detect the malicious documents based on path features?
2. Our methodology detects the malicious links inside the documents that can be used to exploit the user?
3. Our methodology depicts accurate results with area under the curve?

A. Results of experiment 1

The first experiment was carried out using the features discovered through the structure of an unzipped Microsoft Word file. In this experiment we take unzipped files path as features. Python library was used to extract the features and total features that we got were 2964 both from the malicious and benign files. Second phase includes the cleaning and selection of features to avoid complexity and error ration while training of data. Through selection we selected total 84 features and trained upon our designed model.

This experiment provides a good accuracy rate as well as precision rate both for the benign and malicious files as 0.90 and 0.98 respectively. Regardless of this, it gives somehow a low TPR in detection of malicious files of 0.84 but very high TPR in detecting benign files.

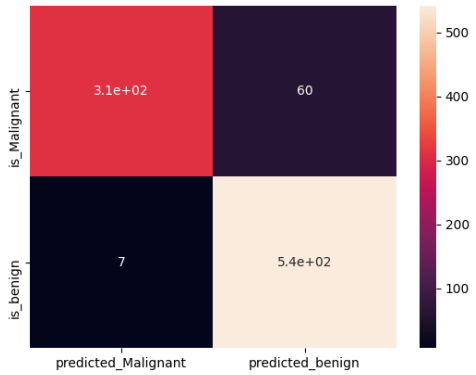

Fig. 8. Confusion matrix for Experiment 1

B. Results of experiment 2

In experiment 2, dataset was downloaded through an online source. We analyze the dataset and extract the lexical features from the URL without visiting the sites to save the system from being injecting. Total feature that is extracted are 22 both from the benign and malicious URL. In the second phase we build an ensemble model just like the experiment 1 with four different machine learning algorithms including random forest, support vector machine, logistic regression and adaboost. We again trained our model

through ensemble learning. The model predicts the URL into three classes that are benign, malicious and malware as 0, 1 and 2 respectively. After applying the test data to the model we achieve the accuracy of 0.99. We achieved a greater accuracy rate in detecting malicious URL using its lexical features. This experiment detects malicious URL but our goal was to detect the malicious documents. So we performed another experiment that detects the malicious documents using the results from this experiment and classify them into benign and malicious.

C. Results of experiment 3

In third experiment we combine the classifiers from both prior experiments and provide the results in combination of these two experiments. We not only provide the ensemble learning but also logical reasoning in order to obtain our required results. Results of Experiment three which detects the malicious documents based on paths features and URL features are as follows. After training of model upon training set we gave the model a testing set to check out the validity of model. We found out that out of 593 benign files only 22 was misclassified as malicious and out of 318 malicious files only 7 files were misclassified as benign, so we got a high recall rate of 98% in detecting malicious Microsoft Word Files.

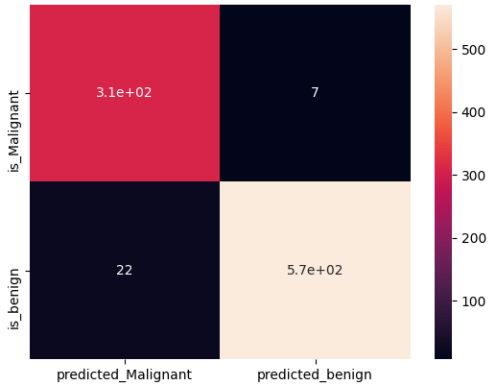

Fig. 9. Experiment 3 Confusion Matrix

The Fig. 9 shows the graphical representation of the confusion matrix. Our model predicts the data with average rates of 0.97 accuracy, 0.97 recall rate, 0.97 precision rate, 0.97 f1-score and 0.97 ROC-AUC score.

Table 2 Classification report of experiment 3

	precision	**Recall**	**f1-score**	**support**
0	0.99	0.96	0.98	593
1	0.93	0.98	0.96	318
Accuracy			0.97	911

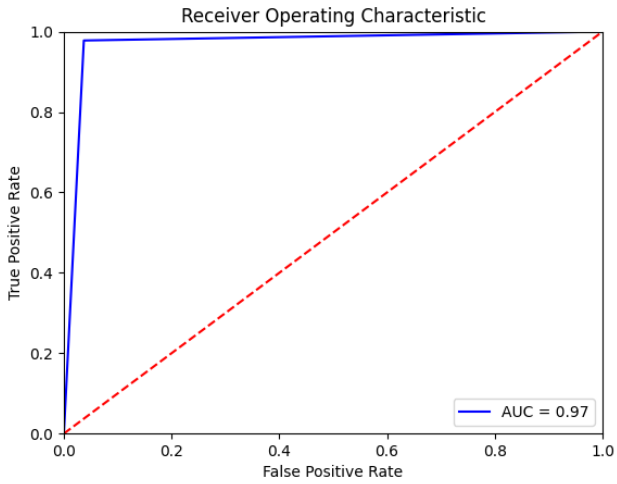

Fig. 10. ROC-AUC curve

We get an ROC-AUC curve over no kill model to be 0.500 and over an ensemble model to be 0.97 approximately.

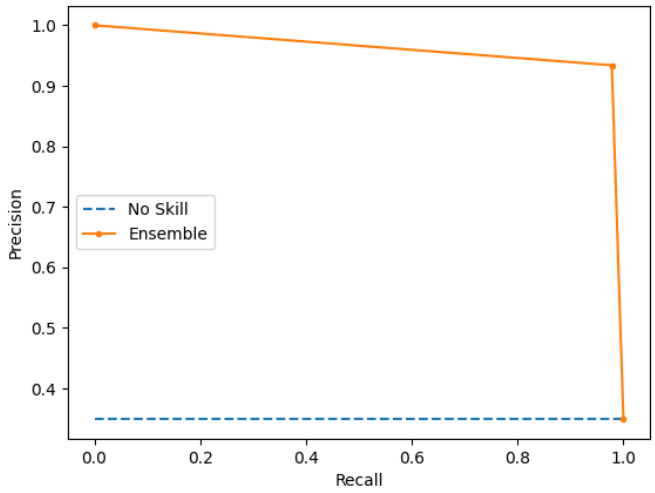

Fig. 11. Precision-Recall plot

The Fig. 11 is showing the curve of precision recall rate with f1 score 0.955 and accuracy score of 0.97 approximately.

V. DISCUSSION AND CONCLUSION

In our thesis we discussed how to analyze and detect the malicious Microsoft Word documents through structural paths of zipped document and through its different external links used inside a document enclosed in tags. We used 3064 Microsoft Word documents with .docx extension among which 1251 were malicious and 1813 were benign files. For further, we used 857 different URL among which 329 were malicious and 528 were benign ones for the training of model for detection of malicious URL from .docx files

We performed structural path feature analysis upon Microsoft word documents. We extracted total 2965 unique path features collectively both from the benign and malicious Microsoft Word documents.

Extracted features were selected through their higher information gain. After preprocessing, we select 84 unique features upon which the model is train[17,18]. We had another dataset of URL with labeled data into benign and malicious. We extracted total 22 lexical features from the URL dataset to train our model. In the next phase, we performed three different experiments depending upon our designed model and achieve our goal that leads us to the TPR of 98%.

Cohen and Nissim [14] used only the structure of Microsoft Word Document as feature and achieve 94.4% TPR while Lu [15] uses the features from multiple views and achieve 97.38% TPR in detecting malicious documents but their model does not detect the malicious links inside the tags whose target is external. Our method uses the dataset that Lu [15] uses and detects not only these malicious links inside the tag but also surplus their results in detecting malicious Microsoft Word files with a TPR of 98%. Our method not only detects the malicious links whose target is some external sites or embedded documents but also detects the macros, embedded objects and other OLE objects that are harmful with average accuracy rate of 0.97 with a TPR of 0.97 and FPR of 0.012 that is the highest from past researches.`

VI. LIMITATIONS

Our proposed methodology depicts the best results as earlier no work has been performed to detect the malicious documents of Microsoft Word by analyzing the embedded URLs. Although our methodology depicts some low TPR in detecting benign files as compared to malicious but in future more work can be done upon the detection of malicious documents based upon the embedded URL inside the tags and structure.

VII. FUTURE WORK

Microsoft Word is the most commonly use document both for business and private use. It is one of the majorly using documents over the internet and in emails. Thus more measures must be taken to secure the OOXML documents and its safe transfer over the internet and from one computer to another. We have applied structural path features and lexical features of URL to our method; in future we are expecting to extend this work and applied on other Open XML formats like xlsx and pptx. And we are also expecting that more work will be performed on this strategy to further improve the accuracy rate and TPR.

References

1. AV-Test. The independent IT security Institute. Available at: https://www.av-test.org/en/statistics/malware/ , 2020.
2. Simson L Garfinkel and James J Migletz. New XML-Based Files Implications for Forensics. IEEE Security & Privacy, 7(2):38–44, 2009.
3. Dataset source. Virustotal. Available at: https://www.virustotal.com , 2020.
4. Chih-Ta Lin, Nai-Jian Wang, Han Xiao, and Claudia Eckert. Feature selection and extraction for malware classification. J. Inf. Sci. Eng., 31(3):965–992, 2015.
5. Sensorstechforum. Popular windows file types used malware. Available at: https://sensorstechforum.com/popular-windows-file-types-used-malware-2018/ , 2018.
6. Aviad Cohen, Nir Nissim, Lior Rokach, and Yuval Elo vici. SFEM: Structural feature extraction methodology for the detection of malicious office documents using machine learning methods. Expert Systems with Applications, 63:324– 343, 2016.
7. Threat Extraction. A preventive method for document-based malware. Available at: https://blog.checkpoint.com/2019/10/10/threatextraction-a-preventive-method-for-document-based-malware/ , 2019.
8. R Vijayakumar, Mamoun Alazab, KP Soman, Prabaharan Poornachandran, and Sita Lakshmi Venkatraman. Robust intelligent malware detection using deep learning. IEEE Access, 7:46717–46738, 2019. doi:10.1109/access.2019.2906934.
9. Michael Darling, Greg Heileman, Gilad Gressel, Aravind Ashok, and Prabaharan Poornachandran. A lexical approach for classifying malicious URLs. In 2015 international conference on high performance computing & simulation (HPCS), pages 195–202. IEEE, 2015.
10. Dataset source. Virus share. Available at: https://virusshare.com , 2020.
11. Dataset source. Contagio. Available at: http://contagiodump.blogspot.com , 2020.
12. Frank Vanhoenshoven, Gonzalo Nápoles, Rafael Falcon, Koen Vanhoof, and Mario Koppen. Detecting malicious URLs using machine learning techniques. In 2016 IEEE Symposium Series on Computational Intelligence (SSCI), pages 1–8. IEEE, 2016.
13. Christophe Chong, Daniel Liu, and Won Hong Lee. Malicious URL detection, 2009. ……
    a. % Reference 13
    b. \bib item[Author6(year)]{ref-journal}
    c. Divya Kapil, Atika Bansal, Anupriya, Nidhi Mehra, Aditya Joshi ,Machine Learning Based Malicious URL Detection. {\em IJEAT} {\bf 2019}, {\em 8}, 2249 – 8958.
14. Nir Nissim, Aviad Cohen, and Yuval Elo vici. Aldocx: detection of unknown malicious Microsoft office documents using designated active learning methods based on new structural feature extraction methodology. IEEE Transactions on Information Forensics and Security, 12(3):631–646, 2016.
15. Xiao feng Lu, Fei Wang, and Zifeng Shu. Malicious word document detection based on multi-view features learning. In 2019 28th International Conference on Computer Communication and Networks (ICCCN), pages 1–6. IEEE, 2019.
16. T. Ngo. Office Open XML Overview, ECMA TC45 white paper. Available at: http://www.ecmainternational.org/news/TC45_current_work/OpenXML%20White%20Paper.pdf , 2011. Accessed: April 14, 2011.
17. Shahid, M. A., Safyan, M., & Pervez, Z. (2024). Improvising the Malware Detection Accuracy in Portable Document Format (PDFs) through Machine Learning Classifiers. *Review of Applied Management and Social Sciences*, *7*(4), 201-221.
18. Shahid, M. A., Iftikhar, M. A., Gondal, Z. A., Adnan, M., & Rathore, S. (2018, December). Object size measurement through images: An application to measuring human foot size. In *2018 International Conference on Frontiers of Information Technology (FIT)* (pp. 298-302). IEEE.