Journal of Information Engineering and Applications
ISSN 2224-5782 (print) ISSN 2225-0506 (online)
Vol.12, No.2, 2022

Malicious Malware Detection Using Machine Learning Perspectives

Temitope Olubunmi Awodiji
Information Securities Emphasis
University of Cumberlands
Williamsburg, KY
tawodiji79374@ucumberlands.edu

Abstract
The opportunity for potential attackers to use more advanced techniques to exploit more people who are online is growing. These methods include getting visitors to click on dangerous URLs that could expose them to spam and ads, financial fraud, defacement of their website, and malware. In this study, we tested different machine learning algorithms against a set of harmful URLs to see how well they worked overall and how well they found malware, spam, defacement, or phishing. The ISXC-URL-2016 dataset from the University of New Brunswick was used to make the dataset. The data was evaluated in Weka using the Random Forest, Decision Tree, Naïve Bayes, and Support Vector Machine algorithms. Each evaluation had a split of 80% of the data and a 5-fold, 10-fold, or 15-fold cross-validation. It was found that the 10-fold Random Forest algorithm correctly categorized 98.8% of the dataset's cases with the most accuracy. The results of this experiment showed that machine learning can be a useful tool for companies that want to improve their security. Despite different limitations encountered in the completion of this research, This study is the most comprehensive available on the use of practices relevant to Malware detection.
Keywords: Machine Learning, URLs, Random Forest, Naive Bayes, Decision Tree, Support Vector Machine
DOI: 10.7176/JIEA/12-2-02
Publication date: November 30th 2022

I. Introduction

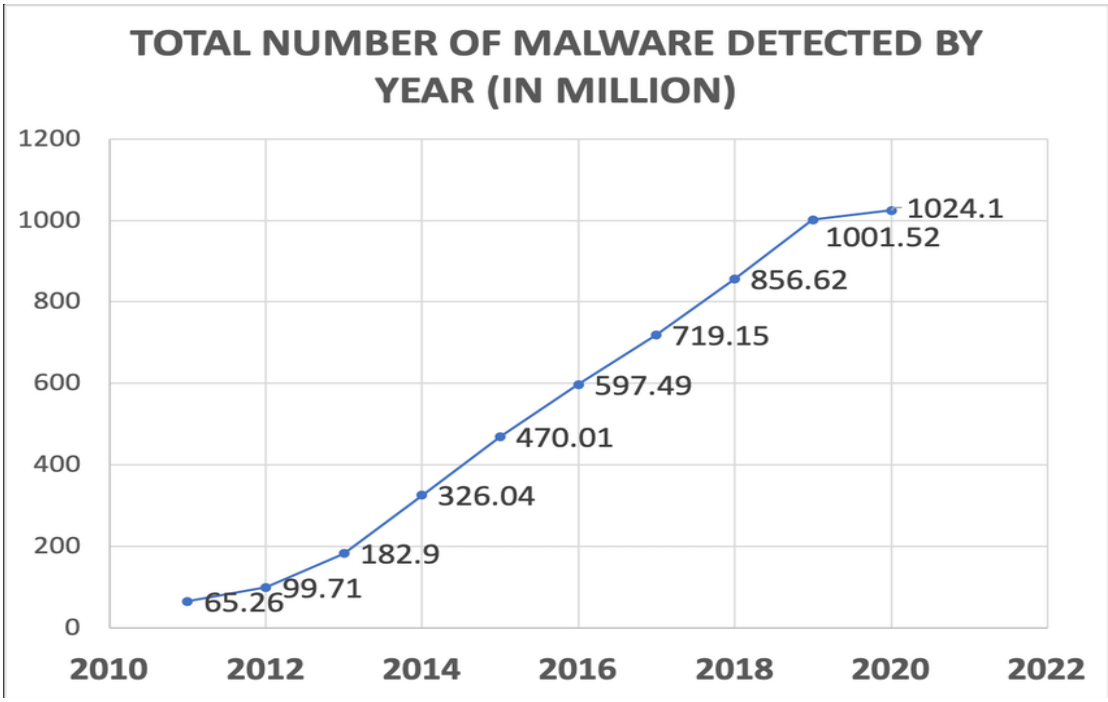
Business, education, research, and access to many essential services in our everyday lives have all changed because of the internet and the World Wide Web. All barriers to communication are removed. Even though the internet was created to encourage learning and connection, some bad characteristics may upset this equilibrium. Among them are money laundering, identity theft, business email breach, malware installation, and other illegal activities. Malicious programs are pieces of code designed to get access to a computer system and obstruct it from functioning normally. Malicious software can enter a computer in a variety of ways. The most common method is to visit rogue websites. The Uniform Resource Locators (URL) of the website one wants to access are frequently entered by users without their knowledge of whether the website is trustworthy or not. Unwanted content is hosted on the rogue website, tempting internet users to fall into the trap. Cybercriminals continue to enhance their wide range of attacks, even during the most recent pandemic, causing massive financial damages. Sometimes malicious code might be found in an email's body text or attachment. The code enters the system and begins automatic execution as soon as users click it.

Users do not know that harmful programs have infiltrated their system until the performance of the system begins to deteriorate, at which point they suspect the presence of malicious programs, but the damage has already been done. The need to identify these URLs arises from the possibility that they could damage systems or steal user data. This need necessitates the idea of using the Machine Learning method to automate the process of classifying URLs.

This paper suggests a mechanism to handle class imbalance and categorize a URL as dangerous or benign. The accuracy of the prediction or classification depends on several variables, including the chosen methods, the features chosen, and the volume of data. It was decided on which features to prioritize, and feature selection was implemented. There are five sections in the paper. The introduction is in Section I, followed by Related Work in Section II, which summarizes previous research in this field. In Section III, all the algorithms utilized to implement the suggested system are thoroughly detailed and research methodology. Results are discussed in Sections IV and V, followed by a conclusion and recommendations for the future.

I.I. RELATED WORK

The below graph shows the Total number of Malware detected by year (in Million). [1]. The graph below shows that malware is increasingly prevalent.



x

The researchers in [2] describe events such as COVID-19 as having played the part in increasing the number of fraudulent websites, coupled with the number of malicious URLs on the internet necessitating a requirement for more advanced detection methods to prevent users from being affected by malware. The researchers of this paper utilized a dataset of malicious URLs with 39 features and analyzed the most optimal set of features and compared various classification methods to determine which would be best at predicting malicious URLs. In the data preprocessing phase, the data first goes through under sampling (decreasing the sample size to balance it with the minority class) to have an even number of benign and malicious URLs, then extracts both lexical and content-based features. They then converted categorical features into numeric features. In the classification phase, the data was compared with Random Forest, Naïve Bayes, LSTM, and CNN. It was found that Naïve Bayes performed the best.

In [3], the researchers describe utilizing ensemble learning to detect malicious URLs. The authors claim that Google's safe browsing report showed that there was a 2800% increase in phishing websites from September 2010 to September 2020. While malicious URLs have been around since the early days of the internet, there is still research to be done on utilizing machine learning to automate detection. The researchers in this paper created the Cyber Threat Intelligence-based Malicious URL Detection Model which gathers the following features: URL-based features, Whois information, and cyber threat intelligence features. The CTI-MURLD model then uses n-gram for feature extraction. Finally, the model utilized several classification algorithms for training. The dataset used was obtained from Kaggle, a compilation of various datasets from Phishtank, and ISXC-URL-2016. The data was split into 70% for training and 30% for testing. The results showed that the Random Forest classification algorithm performed the best.

In [4], the researchers attempt to detect phishing websites using machine learning. Phishing utilizes techniques to lure users into providing sensitive information under the guise of a seemingly legitimate website. Phishing often goes hand in hand with malicious URLs which is why this paper is relevant to our research. The purpose of this paper is to identify malicious URLs from a large dataset using the Recurrent Neural Network approach. Traditional methods include a heuristic approach to classifying phishing websites. Generally, a blacklist and whitelist of phishing websites are maintained. However, the number of phishing sites exponentially increasing negatively affects the performance of the heuristic method. The AlexaRank dataset is used for benign websites and the Phishtank dataset is obtained for phishing sites. The Long Short-Term Memory algorithm was used for training the dataset. The results showed that the LSTM had an average success rate of 86%.

In [5], the researchers described one of the most common methods of combating malicious URLs and drive-by-downloads as utilizing Intrusion Detection/Prevention systems. However, the researchers claim they are limited in design because they are signature-based, meaning that they only can detect/prevent known URLs. There is little to no usefulness when presented with unknown URLs. Therefore, the researchers attempted to classify malicious URLs using a combination of machine learning with natural language processing features. The

researchers used the Support Vector Machine as the machine learning algorithm. Using a dataset consisting of benign URLs and malicious URLs, the researchers categorized a malicious URL by lexical (or properties of the URL itself), host-based (attackers fabricating DNS, WHOIS, and geographical info), URL content-based (inspecting for Active-X or JavaScript), and URL behavior-based (malware from a malicious URL directly affecting the target system). Features of the dataset included length, unsafe extensions, special characters, and the ratio of URL parts. The results showed that, when comparing multiple classification algorithms, the overall results were highly accurate. In conclusion, the researchers noted that real-world applications might have more focus on false positives and false negatives rather than accuracy.

In [6], researchers applied machine learning and data mining concepts to identify malicious websites. The researchers argue that feature engineering is one of the key concepts of machine learning, as the more features there are, the more accurate classification can be. However, the researchers proposed a new method called URL embedding to identify the correlation between various URLs that would help gain more accurate results. The datasets used were the Alexa top 10000 domains for benign URLs and malicious URLs from 360 Netlab. By using standard classification algorithms, the researchers found that URL embedding outperformed traditional feature engineering in each classification algorithm.

In [7] researchers analyzed various adware to classify whether the adware was malicious or not. The researchers used three machine learning algorithms: Random Forest, Naive Bayes and Support Vector Machine (SVM) to see which algorithm performed the best. Random Forest performed the best with an accuracy rate of 0.9947.

I.I.I. Research Methodology

The methodology we utilized to make the prediction is shown below. This framework has been modeled [8] after the paper “COVID-19 Mortality Prediction Using Machine Learning Techniques” [8].

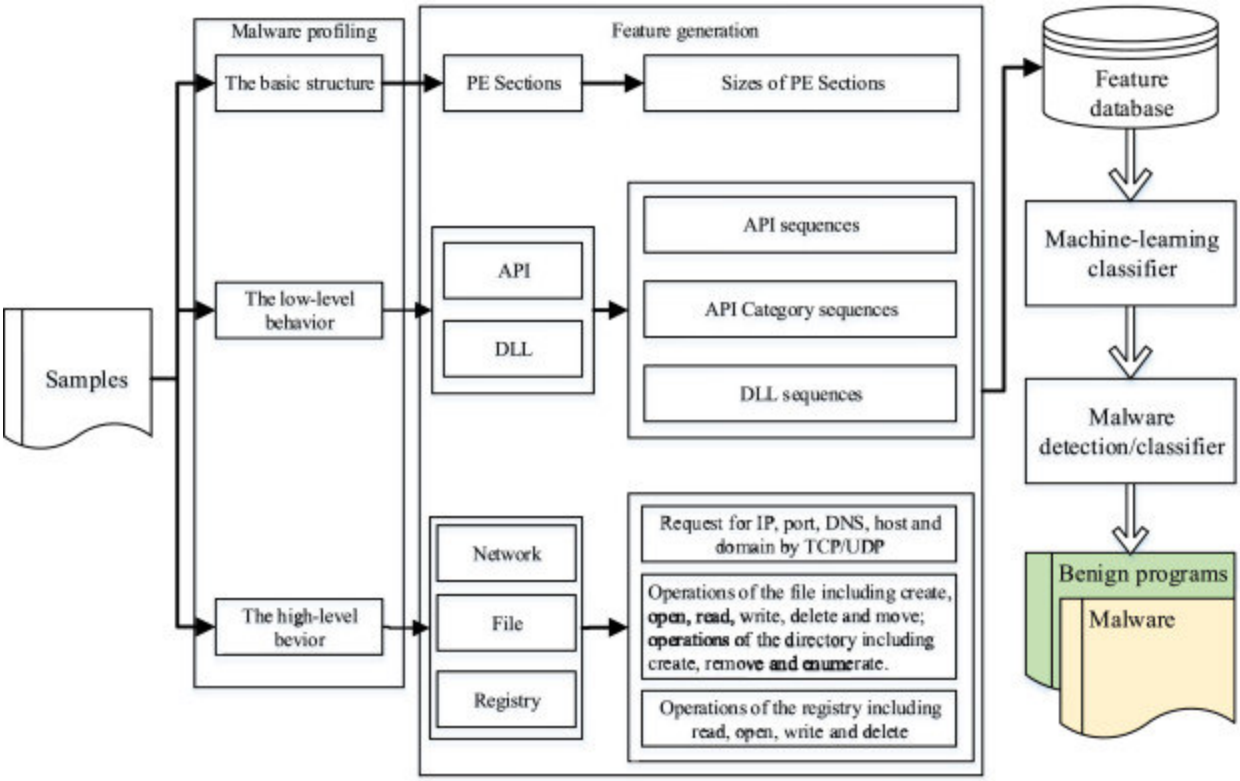
Data collection, data preparation, machine learning, and incremental learning are the four steps that make up this framework. The ‘Data Collection’ stage explains the data gathering process. The data for this research article was taken from the University of New Brunswick data repository [9].

The process of preparing the data for the machine learning algorithm is described in the ‘Data Preparation’ stage. Data cleansing and treatment for missing values are included in this stage.

The "Machine Learning" stage describes the selected models and assesses whether improved accuracy may be achieved through feature selection.

The 'Incremental Learning' stage, the last step, explains which model is appropriate for this use case and indicates whether post-processing is helpful.

The Below diagram shows a Systematic Profiling based malware detection framework [22]



1.1.2 Feature Selection

Feature selection can be used to build various models with different subsets of data. Feature selection helps to improve the performance of the model [10]]. Feature selection helps to reduce the amount of data, resulting in faster training of data. The accuracy usually improves with feature selection, as there is fewer misleading data. Feature selection also helps to reduce overfitting.

Weka's detected 4 features during its feature selection/ranking process. They include URL_sensitiveWord, dld_domain, executable, isPortEighty and ISIpAddressInDomainName.

Two datasets were used to evaluate each model, one with feature selection and the other without. These runs' outcomes were compared for performance and accuracy.

Filter

Nine attributes in the dataset were NaN values. Using the built-in Filter feature of Weka, these attributes were removed. The attributes that had NaN include: avgpathtokenlen, Number Rate_ Directory Name, Number Rate_ File Name, Number Rate_ Extension, Number Rate_ After Path, Entropy_ Directory Name, Entropy_ Filename, Entropy_ Extension and Entropy_ After path.

Numerous fields have successfully used the random forest model. This can be used in modeling various use cases involving regression, classification, and prediction [11]. We used two data sets—one with feature selection and the other without— to run the Random Forest model. Using Weka's built-in feature, missing values were filtered out. To record the accuracy and performance of the Random Forest model, we used Weka's built-in 'Test Option' feature, namely the 5-fold, 10 fold, 15 fold, and 80% split.

Decision Tree (J48) was used in our analysis to determine whether a URL was malicious or not. Decision Tree (J48) is used for classification and performs accurate results of the classification [12]. Decision Tree is a widely used algorithm based on if-then-else statements. It is commonly used in a top-down approach. A selected attribute (root) is split into further nodes (internal) and subsequently split into leaves.

The final classifier, Support Vector Machine, was used in our testing to determine if a given URL is indeed malicious. Recall that Support Vector Machines can categorize new data into separate groups based on existing features. In this case, we are determining if URLs are malicious or do not contain malware. It should be noted that the dataset contains other categories of URLs, but for this purpose, we are identifying any that can contain malware, therefore the rest of the categories will be identified as non-malware. Support Vector Machine can categorize this via a hyperplane in an N-dimensional space. Recall that one of the advantages of Support Vector Machine is that it can handle both linear and non-linear data. This is especially relevant in our dataset, as there is a significant number of features present, of which there is a mixture of linear and non-linear data. However, Support Vector Machine is not suitable for larger datasets. However, in this case, the dataset we are using is relatively small.

In keeping with the same methodology used in previous classification algorithms, Weka was used as the machine learning environment. Weka utilizes John Platt's sequential minimal optimization algorithm as its implementation of Support Vector Machine. Recall earlier that Support Vector Machine is not suitable for larger datasets – this is because it cannot easily solve standard quadratic programming techniques. For example, according to Platt, a dataset containing more than four thousand training examples cannot fit into a 128 Megabyte matrix. Sequential minimal optimization aims to resolve this using chunking, which can break down a large quadratic programming problem into smaller ones [13].

1.1.3 Evaluation and Results

A. When employing the 10-fold test option with the feature selection dataset, the random forest model yielded the highest accuracy of 98.8%. This model ran within 2.06 seconds. The dataset without feature selection had the weakest result, with an accuracy rate of 98.6%. Additionally, this ran slower (2.26 seconds) compared to others.

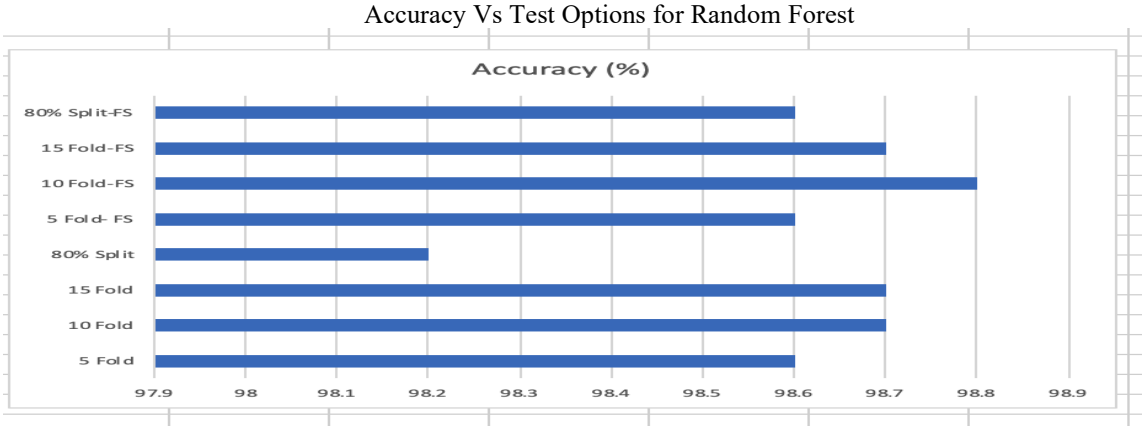
In research studies, the use of a confusion matrix is well-established. The confusion matrices capture true positive, true negative, false positive, and false negative. 1960 instances were identified as malware, 1407 as defacement, 689 as benign, 355 as phishing, and 3768 as spam using Random Forest.

```
=== Confusion Matrix ===

    a    b    c    d    e   <-- classified as
 1407    3    1    2    1 |    a = Defacement
    4  689    9    3    3 |    b = benign
    0    1 1960    9    3 |    c = malware
   15   18    6  355   20 |    d = phishing
    3    0    0    2 3768 |    e = spam
```


The bar graph is plotted with accuracy on the x-axis and Test Options on the Y-axis. The Test options had two broad categories one with feature selection on and the other with no feature selection. The data with feature selection on as suffixed with "-FS".

Accuracy Vs Test Options for Random Forest



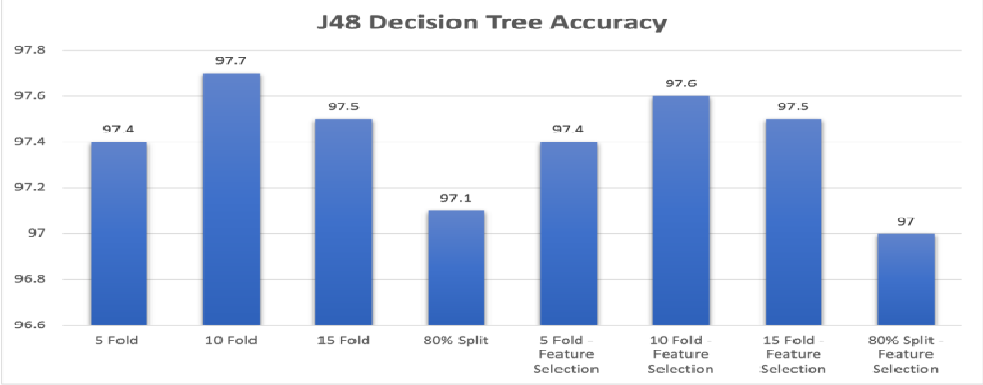
B. Decision Tree

The J48 algorithm was used to determine whether a URL is malicious or not. Weka's implementation of the C4.5 decision tree is the J48 algorithm. A decision tree classifier is what the C4.5 algorithm is[13]. The C4.5 employs a tree-like structure to model decisions, outcomes, and predictions. The decision tree is similar to a flowchart in that it uses a series of if-else decisions to predict the outcome of the data.

In Weka, we used the "test option" feature, which provided 5-fold, 10-fold, 15-fold, and 80% split options. When using the J48 decision tree model to analyze the dataset with full features, the accuracy rates were 97.4%, 97.7%, and 97.5%, respectively. The 80% split resulted in a 97.1% accuracy rate.

When utilizing feature selection in Weka, we removed five attributes. Those attributes were: URL_sensitiveWord, dld_domain, executable, isPortEighty and IsIPAddressInDomainName. The attributes were removed to optimize our dataset. The optimization of the dataset would result in higher classification rates. In our analysis with feature limitation, the 5, 10, and 15-fold yielded accuracy rates of 97.4%, 97.6%, and 97.5% respectively. The 80% split yielded an accuracy rate of 97.0%.

The data shows that using the full dataset with 10-fold validations results in the highest classification rate at 97.7%. The data also shows that utilizing feature selection coupled with an 80% split resulted in the lowest classification rate at 97.0.

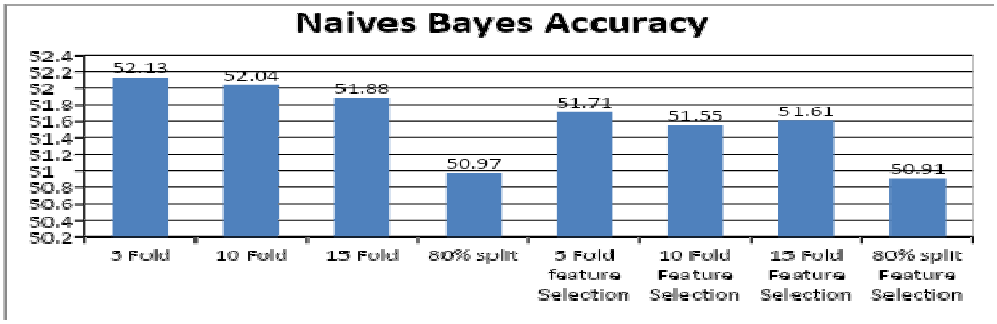


C. Naive Bayes

Naive Bayes was used to determine whether a URL is malicious or not. In Weka, we used the "test option" feature, which provided 5-fold, 10-fold, 15-fold, and 80% split options. When using Naive Bayes to analyze the dataset with full features, the accuracy rates were 52.13%, 52.04%, and 51.88%, respectively. The 80% split resulted in a 50.97% accuracy rate. The average accuracy of Naive Bayes is 53.94%.

When utilizing feature selection in Weka, we removed five attributes. Those attributes were: dld_domain, URL_sensitiveWord, executable, isPortEighty and IsIPAddressInDomainName. The attributes were removed to optimize our dataset. The optimization of the dataset would result in higher classification rates. In our analysis with feature limitation, the 5, 10, and 15-fold yielded accuracy rates of 51.71%, 51.55%, and 51.61% respectively. The 80% split yielded an accuracy rate of 50.91%.

The data shows that using the full dataset with 5-fold validations results in the highest classification rate at 52.13%. The data also shows that utilizing feature selection coupled with an 80% split resulted in the lowest classification rate at 50.91%.



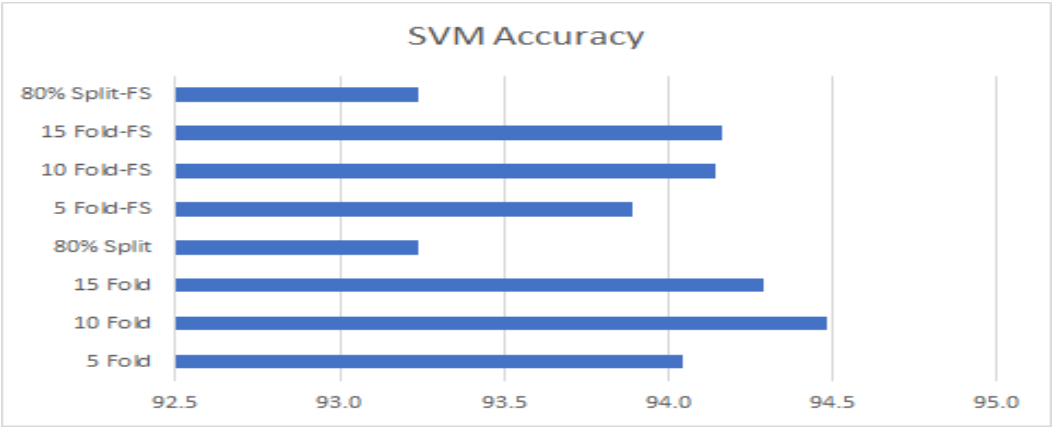
D. Support Vector Machine (SVM)

The first run with SVM with the dataset using Weka was with all features enabled. Running the dataset against Sequential Minimal Optimization with 5-fold cross-validation took a total of 2.45 seconds to build, resulting in 94.0473% of the instances being correctly classified and 5.9527% being incorrectly classified.

Running the dataset again with 10-fold cross-validation took 2.17 seconds to build, resulting in 94.482% of the instances being correctly classified and 5.518% being incorrectly classified. Finally, running the dataset with 15-fold cross-validation took 2.11 seconds to build, resulting in 94.2888% of the instances being correctly classified and 5.7112% being incorrectly classified. Running the dataset through an 80% split took 0.01 seconds to build, resulting in 93.2367% of the instances being correctly classified and 6.7633% being incorrectly classified.

Next, the dataset was evaluated to identify the most optimal features that we theorized to result in higher classification rates. Using the optimized dataset, we ran through the same models as before. The 5-fold model took 2.08 seconds to build and had a correct classification rate of 93.8904% and an incorrect rate of 6.1096%. The 10-fold model took 2.13 seconds to build and had a correct classification rate of 94.1439% and an incorrect rate of 5.8561%. The 15-fold model took 2.13 seconds to build and had a correct classification rate of 94.1681% and an incorrect rate of 5.8319%. Lastly, the 80% split took 0.01 seconds to build with a correct classification rate of 93.2367% and an incorrect classification rate of 6.7633%.

The data shows that using the full dataset with 10-fold cross-validation resulted in the highest correctly classified rate.



I.1.4 CONCLUSION

As the number of sophisticated methods for infecting systems with malware via malicious URLs increases, organizations must be prepared to utilize more efficient countermeasures in order to improve their overall security posture. Therefore, in this paper, we identified four machine learning algorithms that could be used against a dataset of URLs comprising different categories such as malicious, defacing, spam, phishing, or benign, and compared the performance of each algorithm in its ability to correctly classify test data as being malicious. The algorithms present within Weka were used, namely Random Forest, Naïve Bayes, Decision Tree, and Support Vector Machine. The algorithms were first run against the entire dataset using 5-fold, 10-fold, 15-fold, and 80% split. Afterward, the data was optimized to identify the best features and were run against the same methods. It was discovered that Random Forest had the highest percentage of correctly classified instances at 98.8%. The algorithm with the lowest amount of correctly classified instances was Naïve Bayes, at 50.91%. This research did have its limitations, however. First, the dataset used was from 2016, so a more recent and larger dataset should be obtained in future research. Second, the classification algorithms were used with Weka, which provides a satisfactory introduction to machine learning, but dedicated Python algorithms should be used as well to provide more accurate classification rates.

Finally, more research can be done on optimizing features of a malicious URL dataset through methods such as lexical analysis. More researchers from the United States and around the world are welcome to collaborate on optimizing features of a malicious URL dataset.

V. FUTURE APPLICABLE USE

Cyber-attacks and scams can be carried out using malicious websites. Individuals and organizations can be exposed to malicious URLs via emails, text messages, pop-ups, and advertisements. Email accounts can be compromised, phishing campaigns launched, and malware, spyware, and ransomware downloaded if these URLs are clicked or crawled. If the malicious URLs are clicked, the consequences can be disastrous, including financial loss, data theft, and account compromise. The use of machine learning algorithms to detect malware and malicious URLs with high accuracy can help to mitigate the consequences of clicking on a malicious URL.

REFERENCES

[1] Talukder, S. (2020). Tools and techniques for malware detection and analysis. ArXiv Preprint ArXiv:2002.06819.
[2] M. Aljabri, F. Alhaidari, R. . M. A. Mohammad, S. Mirza, D. H. Alhamed, H. S. Altamimi and S. M. BacharChrouf, "An Assessment of Lexical, Network, and Content-Based Features for Detecting Malicious URLs Using Machine Learning and Deep Learning Models," Computational Intelligence and Neuroscience, vol. 2022, pp. 1-14, 2022.
[3] F. A. Ghaleb, M. Alsaedi, F. Saeed, J. Ahmad and M. Alasli, "Cyber Threat Intelligence-Based Malicious URL Detection Model Using Ensemble Learning," Sensors, vol. 22, pp. 1-19, 2022.
[4] A. K. Dutta, "Detecting phishing websites using machine learning technique," PLoS ONE, vol. 16, no. 10, pp. 1-19, 2021.
[5] Q. T. Hai and S. O. Hwang, "Detection of malicious URLs based on word vector representation and ngram," Journal of Intelligent & Fuzzy Systems, vol. 35, no. 6, pp. 5889-5900, 2018.
[6] X. Yan, Y. Xu, B. Cui, S. Zhang, T. Guo and C. Li, "Learning URL Embedding for Malicious Website Detection," IEEE Transactions on Industrial Informatics, vol. 16, no. 10, pp. 6673 - 6681, 2020.
[7] R. P. Cota and D. Zinca, "Comparative Results of Spam Email Detection Using Machine Learning Algorithms," 2022 14th International Conference on Communications (COMM), 2022, pp. 1-5, doi: 10.1109/COMM54429.2022.9817305.
[8] B. Durden, "Using Machine Learning Techniques to Predict RT- PCR Results for COVID-19 Patients," IEEE, 2021.
[9] "University Of New Brunswick," 2016. [Online]. Available: https://www.unb.ca/cic/datasets/url-2016.html.
[10] S. Li, "Genomic Selection in Chinese Holsteins Using Regularized Regression Models for Feature Selection of Whole Genome Sequencing Data," Animals, 2022.
[11]S. Sahu and B. M. Mehtre, "Network intrusion detection system using J48 Decision Tree," 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2015, pp. 2023-2026, doi: 10.1109/ICACCI.2015.7275914.
[12] Y. Wang, "Analysis of Tree Species Suitability for Plantation Forests in Beijing (China) Using an Optimal Random Forest Algorithm," forests, 2022.
[13] J. C. Platt, “Fast Training of Support Vector Machines Using Sequential Minimal Optimization,” in Advances in Kernel Methods - Support Vector Learning , MIT Press, 1998.
[14] Ihya, Rachida&Namir, Abdelwahed&Filali, Sanaa &Aitdaoud, Mohammed &Guerss, Fatima zahra. (2019). J48 algorithms of machine learning for predicting user's the acceptance of an E-orientation systems. SCA '19: Proceedings of the 4th International Conference on Smart City Applications. 1-8. 10.1145/3368756.3368995.
[15] F. A. Ghaleb, M. Alsaedi, F. Saeed, J. Ahmad and M. Alasli, "Cyber Threat Intelligence-Based Malicious URL Detection Model Using Ensemble Learning," Sensors, vol. 22, pp. 1-19, 2022.
[16] A. K. Dutta, "Detecting phishing websites using machine learning technique," PLoS ONE, vol. 16, no. 10, pp. 1-19, 2021.
[17] Q. T. Hai and S. O. Hwang, "Detection of malicious URLs based on word vector representation and ngram," Journal of Intelligent & Fuzzy Systems, vol. 35, no. 6, pp. 5889-5900, 2018.
[18] X. Yan, Y. Xu, B. Cui, S. Zhang, T. Guo and C. Li, "Learning URL Embedding for Malicious Website Detection," IEEE Transactions on Industrial Informatics, vol. 16, no. 10, pp. 6673 - 6681, 2020.
[19] S. Li, "Genomic Selection in Chinese Holsteins Using Regularized Regression Models for Feature Selection of Whole Genome Sequencing Data," Animals, 2022.
[20] M. S. Swetha and G. Sarraf, "Spam Email and Malware Elimination employing various Classification Techniques," 2019 4th International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), 2019, pp. 140-145, doi: 10.1109/RTEICT46194.2019.9016964.

[21] N. A. ALfouzan and N. C, "A Systematic Approach for Malware URL Recognition," 2022 2nd International Conference on Computing and Information Technology (ICCIT), 2022, pp. 325-329, doi: 10.1109/ICCIT52419.2022.9711614.

[22] Han, W., Xue, J., Wang, Y., Liu, Z., & Kong, Z. (2019). MalInsight: A systematic profiling-based malware detection framework. *Journal of Network and Computer Applications*, *125*, 236-250.