Adaptive Machine learning: A Framework for Active Malware Detection

1st Muhammad Aslam
Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education
School of Cybersecurity and Engineering, Wuhan University
Wuhan, China
aslamhayat@whu.edu.cn

2nd Dengpan Ye
Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education
School of Cybersecurity and Engineering, Wuhan University
Wuhan, China
yedp@whu.edu.cn

3nd Muhammad Hanif
Department of Computer Science
COMSATS University Islamabad, Wah Campus
Wahcantt, Pakistan
hanif-cui@ciitwah.edu.pk

4nd Muhammad Asad
Department of Computer Science
Nagoya Institute of Technology
Nagoya, Japan
m.asad@itolab.nitech.ac.jp

2020 16th International Conference on Mobility, Sensing and Networking (MSN) | 978-1-7281-9916-0/20/$31.00 ©2020 IEEE | DOI: 10.1109/MSN50589.2020.00025

***Abstract*—Applications of Machine Learning (ML) algorithms in cybersecurity provide significant performance enhancement over traditional rule-based algorithms. These intelligent cybersecurity solutions demand careful integration of the learning algorithms to develop a significant cyber incident detection system to formulate security analysts' industrial level. The development of advanced malware programs poses a critical threat to cybersecurity systems. Hence, an efficient, robust, and scalable malware recognition module is essential for every cybersecurity product. Conventional Signature-based methods struggle in terms of robustness and effectiveness during malware detection, specifically in the case of zero-day and polymorphic viruses attacks. In this paper, we design an adaptive Machine Learning based active malware detection framework which provides a cybersecurity solution against phishing attacks. The proposed framework utilize ML algorithms in a multilayered feed-forwarding approach to successfully detect the malware by examining the static features of the web pages. The proposed framework successfully extracts the features from the web pages and performs a successful detection process for the phishing attack. In the multilayered feed-forwarding framework, the first layer utilizes Random Forest (RF), Support Vector Machine (SVN), and K-Nearest Neighbor (K-NN) classifiers to build a model for detecting malware from the real-time input. The output of the first layer passes to the Ensemble Voting (EV) algorithm, which accumulates earlier classifiers' performance. At the third layer, adaptive frameworks investigate second layer input data and formulate the phishing detection model. We analyze the proposed framework's performance on three different phishing datasets and validate the higher accuracy rate.**

***Index Terms*—Adaptive Machine Learning, Cybersecurity, Multilayered, Feedforwarding, Malware, Detection.**

I. Introduction

Recently, Verizon Data Breach Investigations Report (DBIR), Symantec Internet Security Threat Report (ISTR), NTT Security Global Threat Intelligence Report, and EY Global Information Security Survey Reports present the experienced phishing attacks every and highlights the cybersecurity threats. Designing an effective malware detection and mitigation system is continuously growing, as the percentage of phishing attacks is rising exponentially. The damaging spectrum of this intensionally designed malware includes the attacks over cloud computing, data centers, content provides, enterprise computer networks, and individual clients. According to the security research report of checkpoints 2018, almost 82% of manufacturers report such attacks in the last few previous years. Furthermore, almost 77% of IT professionals needs better tools and skills to meet today's cybersecurity challenges [1], [2], [3]. This malware executes the attack on the targeted device or server by taking executable code, scripts, active content, and other software. To detect the malware before its execution is a critical task for antimalware programs. Malware detection through standard, signature-based methods are not an effective solution for the zero-day attack-type of the attack on a system exploiting a system fault on the very first when a system fault detected [4], [5], [6].

The conventional way to detect the malware is composed of a black and white list of the programs' signatures. Blacklist is the list of bad program signature which is blocked by the system to be executed. In contrast, the white list consists of

978-1-7281-9916-0/20/$31.00 ©2020 IEEE
DOI 10.1109/MSN50589.2020.00025

the signature of programs that explicitly granted the execution rights while their signatures are listed as bad [7], [8], [9]. In early solutions, the pioneer security measurements rely on manual creation and detection and mitigation rules. This needs a lot of careful feature selection to decode and tackle the systematic malware representation. These solutions need to be stored and utilized in organized manners to get effective antiviral functionality. Implementation of such rules database against identified fingerprints is challenging and time consuming task [10], [11], [12]. Thus, there is a dire need to come up with intelligent solutions that work independently and effectively.

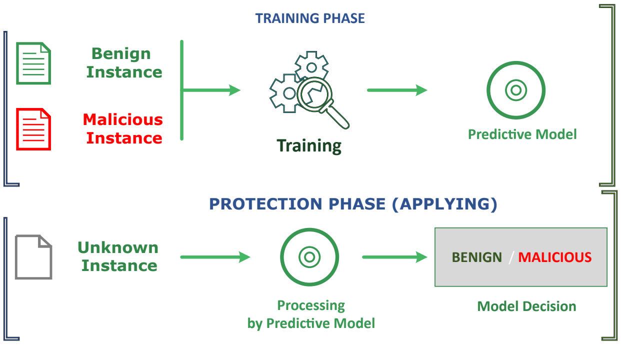

Fig. 1. Machine learning typical system lifecycle

Nowadays, more and more security enabling industrial sectors are adopting machine learning techniques to develop smart filtering and preventions mechanism. Such solutions mainly depend upon the earlier detection and robust reactive mitigation systems. Such a vital integration of machine learning-based detection and mitigation systems provides hundreds of civilian and military-based applications. Smart healthcare is the recent industrial development, and machine learning dependency is on the rise with the passage of every day [13], [14], [15]. Mathematical sophistication of detection and mitigation algorithms of machine learning applications provides promising solutions in the presence of precious datasets citeb17. Similarly, machine learning plays a vital role in both defensive and warfare mode for cyberspace [18]. The conventional setting of machine learning-based malware detection is shown in Fig.1. Machine learning-based products make decisions autonomously and provide a robust way to predict the malware from the data fetched from the vector. One of the drawbacks of ML-based systems for cybersecurity (i.e., malware detection) is when we want to achieve quality, the performance gets affected. Many advanced machine learning algorithms provide decent detection the malware. Such solutions include the Artificial Neural Networks (ANN), Support Vector Machines (SVM), Naive-Bayesian (NB), and Random Forests (RF) [19], [20], [21] [28].

This research aims to introduce an adaptive machine learning framework to keep the proposed design highly sensitive with the feedforwarding technique against phishing attacks. Our proposed multilayered adaptive feedforwarding framework enables the suitable combination of Machine learning algorithms of Random Forest (RF), Support Vector Machine (SVN), and K-Nearest Neighbor (K-NN) classifiers to detect more advanced attacks and work as a counter-adversarial strategy. Adaptive feedforward way of continuous learning for the machine learning algorithms to remain highly reactive with time and new unique data receipt on the system. The way cyberspace is generating data exponentially [29], to cater to the data robustly, the continuous feedforwarding learning model is required to be developed so that it learns from the information on the go [25], [26], [27].

II. Related Work

Many machine learning frameworks caught the considerations of the researcher after earlier success. Multilayered approaches show more relevance for phishing detection and planning preventions. The multilayered deep learning frameworks also promise quick investigation and auto-encode the algorithms in stack fashion and then finalize the classification results after aggregation [9]. eXpose used a Convolutional Neural network (CNN) to classify email and URLs data based on string data. String data is used to extract the low-level features that are further used to train the CNN [11]. Autoencoder outperformed among the SVM and K-NN classification algorithms used on the data extracted from the port-able executable. The executable features were based on the code obfuscation and the Application program interface (API) used during the code [12].

Linear classifiers (i.e., SVM, Logistic Regression (LR)) are more prone to adversarial attacks. In contrast, a non-linear classifier (Markov model, Random forest (RF)) is used as a classifier to detect malware in Android applications. Adversary setup constructed by the author concluded that the non-linear classifiers [13] denies an adversary attack.

DeepAM [14], a custom framework for detecting malware in Pes files and Android application, constructed a heterogeneous in-depth learning approach using the Auto-Encoders (AEs). Support Vector Machines (SVM), C 4.5, and Imbalance data gravitation based approach is used to classify the malware form benign traffics in mo-bile devices. A comparative benchmark prototype system is developed for expanding the limits of the existing system of option for more classifier incorporation while the comparison of result for different classifiers [15], [16].

The research in cybersecurity with the help of machine learning is maturing with time. Due to the evolution of computer technology, both techniques for detecting malware and deceiving the malware system evolve [22], [23], [24]. Adversarial attacks so planned and design that a model's learning capability becomes noising due to noise data's addition to the model learning path. The study optimizes the existing solutions for malware detection with a framework for feedforward prediction and new data to the repository. That model could train with a larger set of data; this will make the model capable of continuous learning to resist the adversarial attacks.

III. Problem Statement

Our goal is to classify a given web page as phishing or not. The problem is formulated as a binary classification task. Consider as set of w webpages $(u_1, y_1)(u_2, y_2), ..., (u_w, y_w)$ where u_w for $w = 1, 2, 3, \ldots.., W$ represents a Webpage and $y_w \epsilon [-1, 1]$ represents the label of webpage instance with $y_w = +1$ being a Phishing Webpage and $y_w = -1$ being a benign webpage. The classification procedure is to obtain a feature representation

$$u_w \rightarrow x_w where x_w \epsilon \mathbb{R}^n \tag{1}$$

where the n dimensional feature vector representation Webpage u_w. The next step is to learn a prediction function $f : R^n \rightarrow R$ which is the score predict-ing the class assignment for webpage instance x The prediction made by the function f the can minimize the total number of mistakes $(\sum_{t=1}^{w} t\mathbb{Q}_{yw<>yw})$ in the entire dataset.

IV. Methodology and Proposed Solution Adaptive Machine learning

Antiviruses use various methods for detecting malware, such as signature-based and heuristic-based techniques. Polymorphic and metamorphic malware employ obfuscation techniques to bypass traditional detection methods used by antiviruses. The recognition process of malware comprises two execution phases; pre-execution phase one determines the nature of the file without executing it. The post-execution phase surveillance the activity of file execution and marks its malware based on data collected during execution. To detect the malware in time will allow the defending system to take necessary action before it activates and starts harming the target system. The conventional way to detect the malware was based on a signature database, which is memory, processor, and network intensive. The traditional algorithm was not so robust and effective in offline mode. With the introduction of machine learning in cybersecurity, cyber defense lines have become harder. Adversarial attacks are one of the kinds of attacks that is a bottleneck for modern machine learning-based solutions in cybersecurity. A framework is developed for continuous learning of model and resistant for adversarial attacks to address the challenge. Initially, the static data is passed through the three classification algorithms of Random forest (RF), Support Vector Machine (SVM), and K-Nearest Neighbor (K-NN). The output of these models is boosted through the ensemble learning algorithm of Ensemble Voting (EV). The EV model is used to predict the test data and further in the framework for the continuous learning process.

A. Detection and Adaptive strategy

The proposed study has novelty in terms of model training's adaptation technique and updating the model without relaying the outer world. As data growth in the cyber world is higher than the power of processing it, we have to innovate a solution that lessor relies on processing time. As the extracting data, feature engineering, and training model is time taking task. Hence, we make an effort to solve the re-training of the model from zero on each new malware appearance. The framework comprises four modules, Feature Extraction and selection, model Training and Performance Improvement, prediction and data storage.

1) Model Training and Performance Improvement: To keep the model up to date, the machine learning model is greedy of data; the model update and training module are responsible for the following action to update the model:

- Sensing of Low Detection Rate/increase of False Positive Rate/decrease in Confidence
- Re-Training of the model on the available data
- Verification of the model performance with Active Model
- Update the active model for detection
- Drop the old model form system
- Seek for next event

2) Prediction: The prediction module provides the interface for provision of data to the model. After processing the data prediction is made in the form of Ture/False.

3) Data storage: The prediction module's prediction and the extracted features are sent to the data storage, which will act as the data source for the next iteration of the model training. All the above four modules serve as the adaptive active malware detection framework. The framework makes the training process automatic and smooth for continuous learning.

B. Dataset

Large datasets are required to train the model, as cyberspace is growing exponentially; hence, the probability of creating new malware is also higher, and growth in malware files also increases, respectively. In this paper, to train and test our proposed framework, we use the following vital datasets; (i) Dataset of static webpages [27], (ii) dataset gathered by PhishTank website [30], and (iii) Phishing dataset of Center for Machine Learning and Intelligent Systems [32]. Meanwhile, the nature of the first dataset [27] is different from the rest of the datasets used in the experiment sections. This dataset contains the subset of normal webpages and attacked webpages from phishing attacks. The features classes of datasets of [27] and [30] are shown in Fig.2 and Fig.3. General characteristics of these datasets contain an email from hundreds of organizations; most organizations came from randomly sampling enterprises that had reports of lateral phishing. Different organizations have a different magnitude of user accounts and email service's nature according to the organization's nature. These datasets share the general schema of office 365 for their email provider and, at a higher level, have different objects within each encapsulated email content. Usually, these objects include the metadata (SPTP identifier), Office 365 identifier, which describes the timestamp, source, destination, subject, and email body. All these email messages have a standard full HTML format.

C. Feature Extraction, Labeling and Selection

The feature selection module is responsible for extracting the features from the raw data of the webpage. The parts

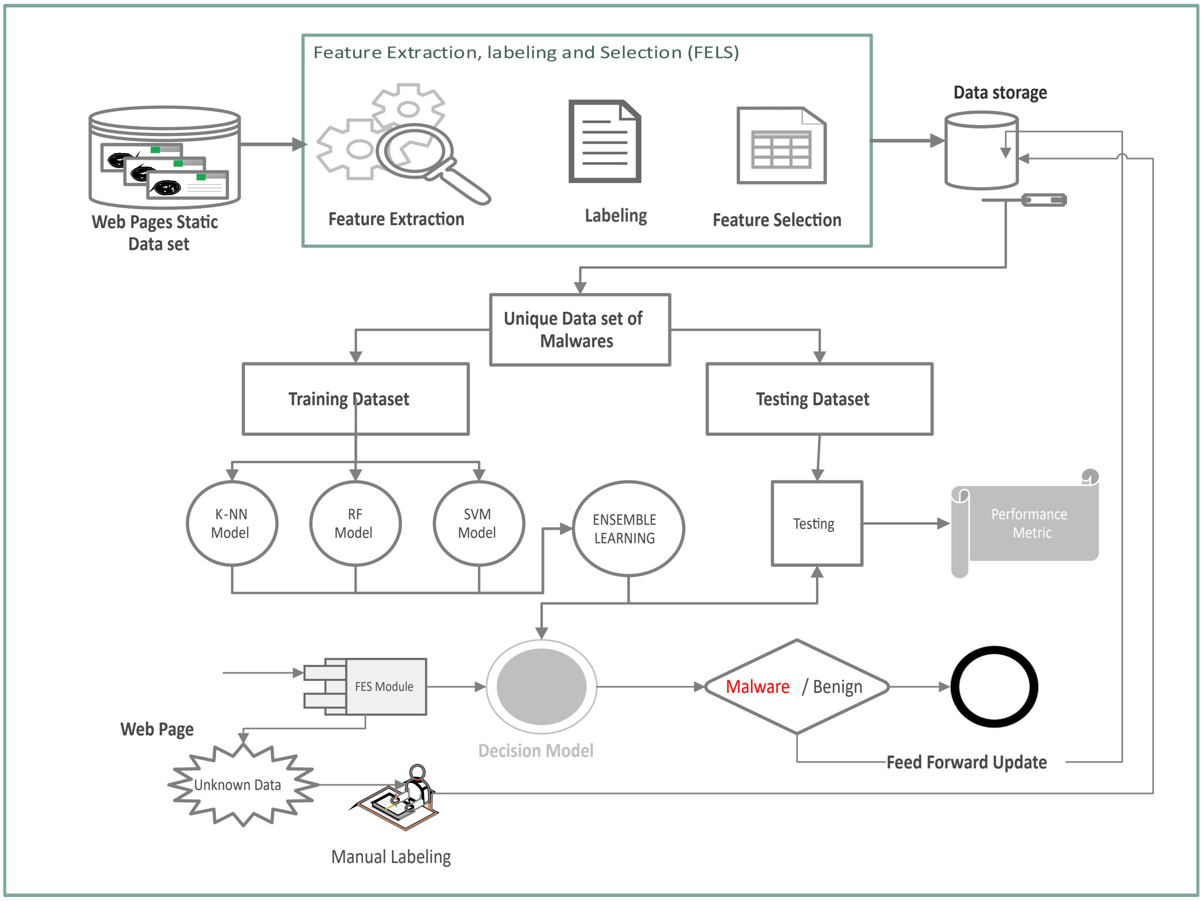

Fig. 2. Malware Detection System Framework overview

are precalculated attributes of the page, which are standard in packet capturing techniques. The pages which are not adequately marked by the machine learning algorithm are kept for manual tagging. The features extraction, labeling, and selection are based on simple routines with some math functions. In this study, data was composed of features selected from five domains: Address bar based features, Abnormal based features, HTM and JavaScript-based features, Domain-based feature, and statistical report based features. Extracted features are then normalized and labeled based on extracted features. Suitable features are selected for the storage of the database for further use of machine learning algorithm processing. Stored Dataset is passed to the module where data is split into two datasets one for training purposes other is testing the prediction model. Splitting is done on 60 to 40 proportion. The training dataset is passed to three hyperparameter-based algorithms from different families of classification; these are Random forest (RF), K-Nearest Neighbor (K-NN), and Support vector machine (SVM) and evaluated the performance measures on each. A feedforwarding link is the key attribute of the proposed framework, which will increase the model's learning with time in running state. The prediction results from the model are fed to the training data repository which contain unique in-stance of features.

D. Machine Learning Algorithms

Machine learning algorithms have a pivot role in adaptive malware detection. The se-lection of the algorithm is tricky. Several algorithms are available; each algorithm has unique features and applications in a specific area. Therefore, it is at the researcher's disposal to choose the best of best to get adequate results. Our proposed framework has a module of training machine learning models with an ensemble of three best performer machine learning algorithms. Subsequent lines briefly introduce those algorithms with their performance. The proposed feedforward multilayered arrangement of the adaptive machine learning framework is shown in Fig.2.

$$p(h \mid d)=\frac{p(d \mid h)*p(h)}{p(d)} \tag{2}$$

Where $p(h \mid d)$ is the probability of the hypothesis, known as posterior probability, $p(d \mid h)$ likelihood of truth. The primary purpose is to achieve the maximum probable view. Gaussian Naive Bayse is the extension of Naive Bayse binary classification based on the rules of Gaussian distribution.

1) Random forest (RF): Random Forests (RF) is also a type of classification algorithm derived from the decision tree group. The operation of RF involves constructing decision trees during the training time and producing the classes (classification) or mean prediction (regression) for each decision tree. Besides, the random decision forests the decision trees precisely by overfitting the training datasets. In particular, RF supports the hyper-parameter, which adaptively tunes to suitable parameters that produce the test results' maximum accuracy.

$$Accuracy = Max \left\{ RF(\sum_{0}^{30}) random_{state} \right\} \tag{3}$$

2) Support Vector Machine (SVM): The Support Vector Machine (SVM) algorithm aims to find a hyperplane in the N-dimensional space $(N|thenumberoffeatures)$ that distinctly classifies the data points. The $SVM-3$ can be divided into three families into the linear SVM, the polynomial SVM, and the RBF SVM. In our model, we used the linear family of the SVM algorithm.

$$Accuracy = Max \left\{ SVM(\sum_{0}^{30} C) \right\} \tag{4}$$

SVM was training for the linear kernel, and the C (Penalty parameter C of the error term) was tuned for best rest results, and we get the best work of 92.69% accuracy while the value of C=5.0.

3) K-Nearest Neighbor (K-NN): The k-nearest neighbor algorithm is a type of classification algorithm derived from supervised learning. This classification algorithm adopts the labeled points and trains them to label the remaining issues. In particular, the labeled end finds the closest point and gets the votes from these neighboring points. Hence, the highest number of neighbors having the same label will be selected as a new label point, where k represents the number of neighbors check by the individual point.

4) Ensemble Voting (EV): The Ensemble Model is a voting classifier model that combines multiple different models, i.e., sub-estimators, into a single model (ideally) more substantial than any individual models. During the construction of the model, the ensemble voting classifier sets the voting to "Hard." The reason behind this, the hard franchise (also known as majority voting) allows every individual classifier to vote for a class, where the majority wins. Each classifier provides a probability value in a soft poll that a specific data point belongs to a particular target class. In our proposed framework, EV accumulates the decision from the output of Random Forest (RF), Support Vector Machine (SVN), K-Nearest Neighbor (K-NN), and classifiers. By default, EV accepts the majority of the verdicts over phishing and other anomaly detection decisions.

E. Cybersecurity Challenges

Major research challenge of final Ensemble Voting (EV) model and other first layer models is to keep the following performance metric satisfied.

1) Attaining Accurate Rate of predictions: The number of False Positives Rate (FPR) results is the indicators of the algorithm's correctness. The false-positive rate tends to rise as algorithms calculate errors and measure the infected label as a benign file. Similarly, The accuracy of True Positive Rate (TPR), False Negative Rate (FNR), True Negative Rate (FNR) is also a primary adaptive machine learning task of the proposed framework. Due to the colossal dataset and massive network traffic, the designed algorithms tend to make mistakes among hundreds of similar files and accepted features [20]. It becomes essential to carefully plan and implement machine learning algorithms and check the particular algorithms' suitability in specific data streams. Most importantly, it is necessary to maintain the robust correctness of the algorithms [21].

2) Identification framework: The detection framework is an extension of the existing machine learning modeling mechanism with forwarding feed update to keep the model current with the model's True Positive prediction. The framework uses static data for training purposes and while real-time data could be predicted from the model.

3) Interpretable Model: As the machine learning algorithms are being implemented with dedicated systems like SVM and provide the users' flexility. Users tend to rely on simple input and output from the system. In these situations, troubleshooting becomes almost impossible for the implementing team. That is why it is critical to keep the design and technical detail in an exact blueprint so the implementing organization can understand the adaptive machine learning algorithms' technical schemes.

V. Results and discussion

The setup for executing the data and algorithm is carried out on python3 with anaconda 3.7.3 version using the web interface of Jupyter notebook 5.7.8 version. In the following sections, the result will be presented, and the performance metric would be discussed. To our adaptive machine learning framework, use the following dataset; dataset gathered by (i) Dataset of static webpages [27], (ii) dataset collected by PhishTank website [30], and (iii) Phishing dataset of Center for Machine Learning and Intelligent Systems [32].

1) Performance metrics: Performance metrics on the bases of which the efficiency will be declared are discussed in the subsequent section of this study:

- Confusion matrix: The confusion matrix is one of the most intuitive metrics and utilizes for finding the correctness and accuracy of the model. It is used for the classification problem where the output can be of two or more types of classes. We are interested in false-positive rate (FPR) [22].
- Accuracy: Accuracy in classification problems and ensures the intend of machine learning implementation and primary measurement of performance metric.
- Precision: Precision is a measure that tells us what proportion of prediction as value true, to actual it is true.

Precision is the ability of the classifier not to label as positive a sample that is negative.

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

- Recall: The recall is the ability of the classifier to find all the positive samples.

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

- F-measure: The F-measure can be interpreted as a weighted harmonic mean of precision and recall. It a good measure which takes precision and recalls into consideration at the same time.

$$F - Measure = 2 \times \left(\frac{Precision \times Recall}{Precision + Recall} \right) \tag{7}$$

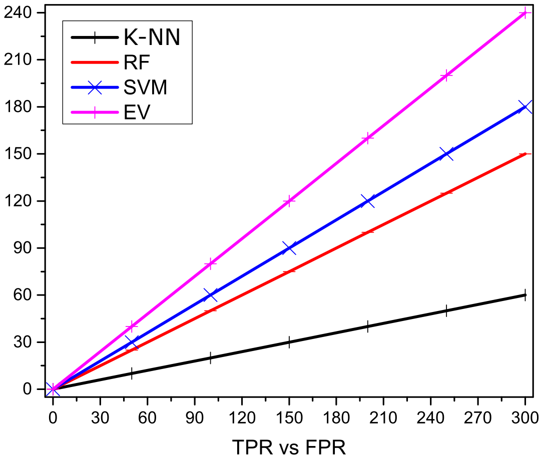

Fig. 3. False Positive Rate versus True Positive Rate

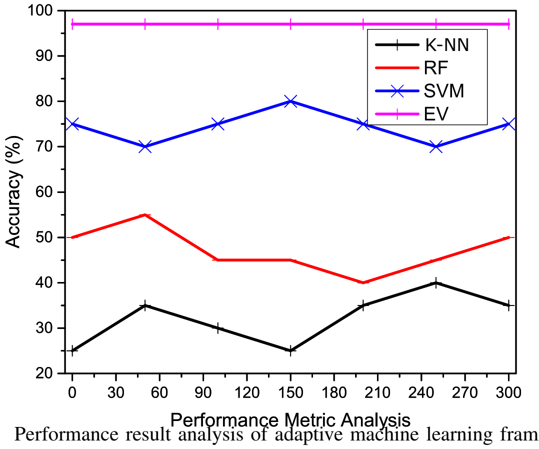

Fig. 4. Performance result analysis of adaptive machine learning framework for [27] static webpage dataset

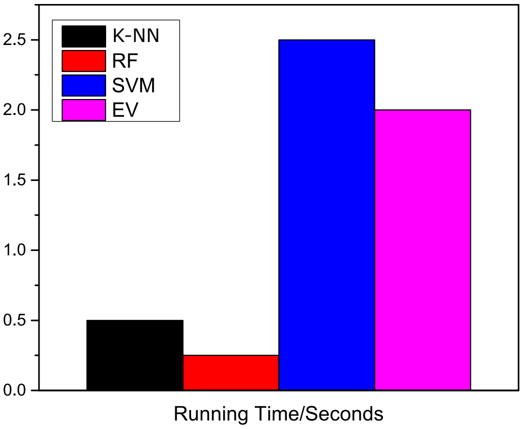

Fig. 5. Simulation computational runtime

2) Results Analysis: Our experimental assessment results are productive because of performance metrics. Our proposed Active Malware Detection is demonstrated based on inspecting our proposed algorithms with Precision, recall, and F-measure metrics. In previous works, the attacks like phishing attacks were detected using the conventual methods, which are now ineffective due to the emergence of computer technologies (i.e., Machine learning, Deep learning). The methodology, whereas we introduce a modern way to train the model for possible future attacks through feedforwarding technique, which will provide resistance to adversarial attacks on the machine learning model, one of the machine learning model problems working in a real-time environment.

Fig.3 is the plotting of true positives out of the positives (TPR = true positive rate) vs. the fraction of false positives out of the negatives (FPR = false positive rate). TPR is also known as sensitivity, and FPR is one minus the specificity or true negative rate. Ensemble voting has a higher Area under the curve, while K-NN has the lowest. In these results, the Ensemble Voting algorithm performs at all other in the analysis and adds up the accuracy at a greater level while minimizing the algorithms' run-time. While its higher running time, which is resource-intensive compared to other algorithms, is our case study, specifically the run time Analysis. An algorithm is worth producing an accurate result, but efficiency is essential for that algorithm's competence. Fig.6 shows the performance of classifiers used in our adaptive machine learning framework in terms of time taken for training and testing the Random Forest (RF), Support Vector Machine (SVN), K-Nearest Neighbor (K-NN), and Ensemble Voting (EV) classifiers. Fig.7 indicates the results of the classification performance of adaptive machine learning algorithms utilize at layer one and layer 2. This graph provides results of the adaptive machine learning accuracy rate of all machine learning algorithms. We can see from the figure that layer 1 provides the initial classification advantage that the EV at the second layer can successfully raise the proposed model's accuracy. This higher accuracy at the second layer becomes the third layer's input, which successfully separates the regular traffic from the perishing attacks. In this result, we consider the accuracy mean and accuracy of predictions. RF performs very effectively at the first layer and achieves the highest accuracy rate, while K-NN and SVM struggle compared to other algorithms. Most importantly, EV archives the 97.6% accuracy mean and 96.42% prediction accuracy.

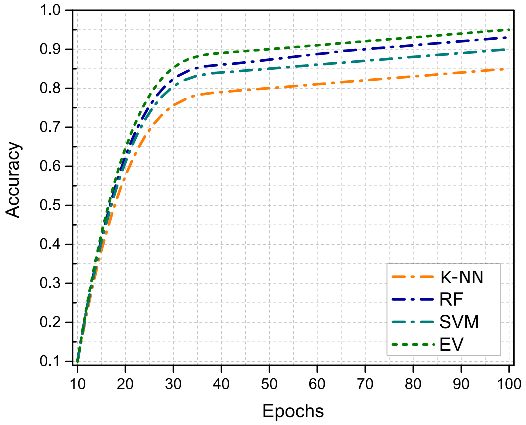

Fig. 6. Malware Detection Accuracy for [27] dataset

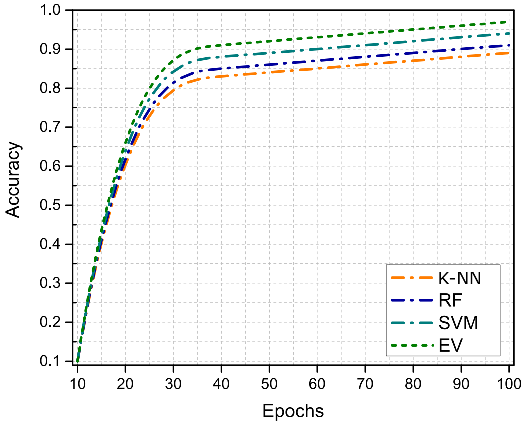

Fig. 7. Malware Detection Accuracy for PhishTank websitet [30] dataset

Fig. 9 shows the final result of an EV based decision of the proposed adaptive machine learning framework for the dataset given by [27]. The output is being successfully classified into benign, malware, phishing, defacement, and spams at a significant accuracy rate. Similarly, Fig. 10 shows the final output of the proposed adaptive framework and classify the input of [30] dataset into two possible outcomes as +1 as regular traffic and -1 as phishing attacks detected successfully. Fig. 11 indicates the final decision results into three classes, in which value 1 represents the regular traffic, 0 as spams traffic, and -1 as detected phishing attacks.

VI. Conclusion and Future work

This research paper proposed an adaptive machine learning framework that consolidates the fact that our freeloading machine learning technique significantly improves accuracy in detecting phishing attacks. Our multilayered approach utilizes the K-Nearest Neighbor (K-NN), Random forest (RF), and Support Vector Machine (SVM) algorithms at the first layer. The first layer's output acts as input for the second layer, where the practical ensemble voting classifier approach boosts the standard classification models. Our research's significant contribution is to design a framework that enhances the machine learning model's performance and enhances the capability of learning by feeding the result to the data repository for later used for training the model. We have utilized four different network traffic data datasets to deal with the dynamic nature of phishing attacks. The proposed system obtained a proficient results-based analysis of performance metrics that detects the malware's utmost precision of 96.1%, Recall of 97.79%, and F-measure of 96.59%. Decision-maker model of EV archives the 97.6% accuracy mean and 96.42% prediction accuracy for detecting phishing attacks.

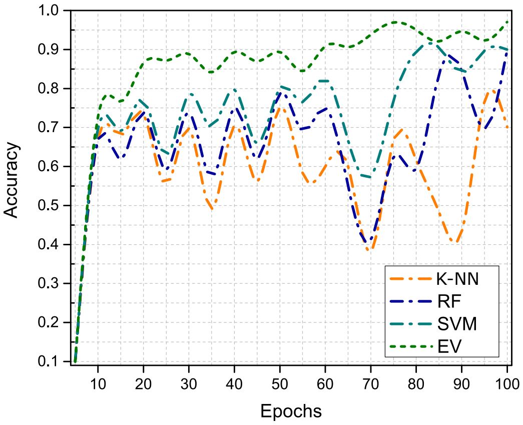

Fig. 8. Malware Detection Accuracy for publicly available phishing Center for Machine Learning and Intelligent Systems [32] dataset

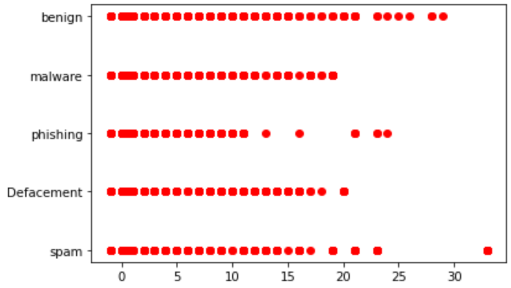

Fig. 9. Phishing attack detection for [27] dataset

VII. Acknowledgement

This research was funded by National Key Research Development Program of China (2019QY(Y)0206), and the National Natural Science Foundation of China NSFC (62072343, U1736211).

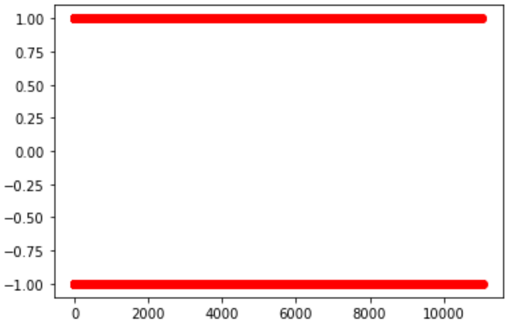

Fig. 10. Phishing attack detection for [30] Phishtank dataset

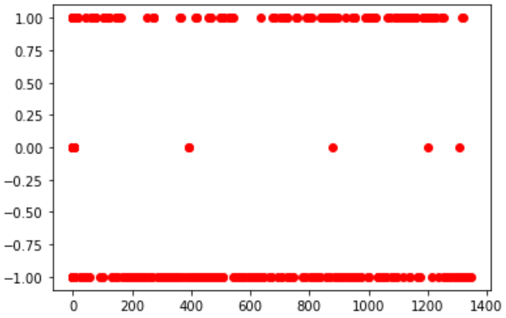

Fig. 11. Phishing attack detection for [32] Phishtank dataset

REFERENCES

[1] Jartelius, M. (2020). The 2020 Data Breach Investigations Report–a CSO's perspective. Network Security, 2020(7), 9-12.

[2] Nitsch, Holger, Julio Hernandez-Castro, Edward Cartwright, Anna Stepanova, and Darren Hurley-Smith. "RAMSES: Internet Forensic Platform for Tracking the Money Flow of Financially-Motivated Malware and Ransomware." European Law Enforcement Research Bulletin 4 SCE (2019): 141-146.

[3] Kaspersky, "Machine Learning for Malware Detection By Kaspersky for Business," 2019.

[4] D. Ucci, L. Aniello, and R. Baldoni, "Survey of machine learning techniques for malware analysis," Comput. Secur., vol. 81, pp. 123–147, Mar. 2019.

[5] Q. Liu, P. Li, W. Zhao, W. Cai, S. Yu, and V. C. M. Leung, "A Survey on Security Threats and Defensive Techniques of Machine Learning: A Data Driven View," IEEE Access, vol. 6, pp. 12103–12117, 2018.

[6] R. M. Yadav, "Effective analysis of malware detection in cloud computing," Comput. Secur., vol. 83, pp. 14–21, Jun. 2019.

[7] A. Calleja, A. Martín, H. D. Menéndez, J. Tapiador, and D. Clark, "Picking on the family: Disrupting android malware triage by forcing misclassification," Expert Syst. Appl., vol. 95, pp. 113–126, Apr. 2018.

[8] S. Rota Buló et al., "Randomized prediction games for adversarial machine learning," IEEE Trans. Neural Networks Learn. Syst., vol. 28, no. 11, pp. 2466–2478, Nov. 2017.

[9] Y. Wang, W. Cai, and P. Wei, "A deep learning approach for detecting malicious JavaScript code," Secur. Commun. Networks, vol. 9, no. 11, pp. 1520–1534, Jul. 2016.

[10] C. Woo, "Using Spam Filters to Detect Malware: A Machine Learning Approach to Malware Detection."

[11] K. B. Joshua Saxe, "eXpose: A Character-Level ConvolutionalNeural Network with Embeddings For DetectingMalicious URLs, File Paths and Registry Keys," arxiv: 1702.08568v1, 2017.

[12] H. Sayadi, N. Patel, S. M. P D, A. Sasan, S. Rafatirad, and H. Homayoun, "Ensemble Learning for Effective Run-Time Hardware-Based Malware Detection: A Comprehensive Analysis and Classification," in 2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC), 2018, pp. 1–6.

[13] Z. Abaid, "Time-Sensitive Prediction of Malware Attacks and Analysis of Machine-Learning Classifiers in Adversarial Settings."

[14] Y. Ye, L. Chen, S. Hou, W. Hardy, and X. Li, "DeepAM: a heterogeneous deep learning framework for intelligent malware detection," Knowl. Inf. Syst., vol. 54, no. 2, pp. 265–285, Feb. 2018.

[15] Z. Chen et al., "Machine learning based mobile malware detection using highly imbalanced network traffic," Inf. Sci. (Ny)., vol. 433–434, pp. 346–364, Apr. 2018.

[16] E. B. Karbab and M. Debbabi, "MalDy: Portable, data-driven malware detection using natural language processing and machine learning techniques on behavioral analysis reports," Digit. Investig., vol. 28, pp. S77–S87, Apr. 2019.

[17] G. Suarez-Tangil, S. K. Dash, M. Ahmadi, J. Kinder, G. Giacinto, and L. Cavallaro, "DroidSieve: Fast and accurate classification of obfuscated android malware," in CODASPY 2017 - Proceedings of the 7th ACM Conference on Data and Application Security and Privacy, 2017, pp. 309–320.

[18] I. Indyk and M. Zabarankin, "Adversarial and counter-adversarial support vector machines," Neurocomputing, vol. 356, pp. 1–8, Sep. 2019.

[19] M. Lichman, "UCI Machine Learning Repository [http://archive.ics.uci.edu/ml].," UCI Machine Learning Repository. Irvine, CA, USA, p. 2013, 2013.

[20] Ondrej Kubovič, "Machine-Learning Era in Cybersecurity: a Step Towards a Safer World or the Brink of Chaos? Tlp: White Machine-Learning Era in Cybersecurity // a Step Towards a Safer World or the Brink of Chaos?"

[21] Z. Chiba, N. Abghour, K. Moussaid, A. El Omri, and M. Rida, "A novel architecture combined with optimal parameters for back propagation neural networks applied to anomaly network intrusion detection," Comput. Secur., vol. 75, pp. 36–58, 2018.

[22] M. Sunasra, "Performance Metrics for Classification problems in Machine Learning," 2017-11-11, 2017.

[23] Li, Jiaqi, Zhifeng Zhao, and Rongpeng Li. "Machine learning-based IDS for software-defined 5G network." IET Networks 7.2 (2017): 53-60.

[24] Tsai, Chih-Fong, Yu-Feng Hsu, Chia-Ying Lin, and Wei-Yang Lin. "Intrusion detection by machine learning: A review." expert systems with applications 36, no. 10 (2009): 11994-12000.

[25] Wang, Zheng. "Deep learning-based intrusion detection with adversaries." IEEE Access 6 (2018): 38367-38384.

[26] Otoum, Safa, Burak Kantarci, and Hussein T. Mouftah. "On the feasibility of deep learning in sensor network intrusion detection." IEEE Networking Letters 1, no. 2 (2019): 68-71.

[27] Hoang, Xuan Dau. "A website defacement detection method based on machine learning techniques." In Proceedings of the Ninth International Symposium on Information and Communication Technology, pp. 443-448. 2018.

[28] Mishra, Preeti, Vijay Varadharajan, Uday Tupakula, and Emmanuel S. Pilli. "A detailed investigation and analysis of using machine learning techniques for intrusion detection." IEEE Communications Surveys Tutorials 21, no. 1 (2018): 686-728.

[29] Wang, Yu, Weizhi Meng, Wenjuan Li, Zhe Liu, Yang Liu, and Hanxiao Xue. "Adaptive machine learning-based alarm reduction via edge computing for distributed intrusion detection systems." Concurrency and Computation: Practice and Experience 31, no. 19 (2019): e5101.

[30] Abdulhammed, Razan, Miad Faezipour, Abdelshakour Abuzneid, and Arafat AbuMallouh. "Deep and machine learning approaches for anomaly-based intrusion detection of imbalanced network traffic." IEEE sensors letters 3, no. 1 (2018): 1-4.

[31] Moore, Tyler, and Richard Clayton. "Evaluating the wisdom of crowds in assessing phishing websites." In International Conference on Financial Cryptography and Data Security, pp. 16-30. Springer, Berlin, Heidelberg, 2008.

[32] Karabatak, Murat, and Twana Mustafa. "Performance comparison of classifiers on reduced phishing website dataset." In 2018 6th International Symposium on Digital Forensic and Security (ISDFS), pp. 1-5. IEEE, 2018.

[33] Mohammad, Rami M., Fadi Thabtah, and Lee McCluskey. "An assessment of features related to phishing websites using an automated technique." In 2012 International Conference for Internet Technology and Secured Transactions, pp. 492-497. IEEE, 2012.

[34] Hoang, Xuan Dau, and Ngoc Tuong Nguyen. "Detecting Website Defacements Based on Machine Learning Techniques and Attack Signatures." Computers 8, no. 2 (2019): 35.