RES MILITARIS
REVUE EUROPEENNE D ETUDES EUROPEAN JOURNAL OF MILITARY STUDIES
Social Science Journal

ENHANCING MALWARE CLASSIFICATION THROUGH FEATURE INTEGRATION IN MACHINE AND DEEP LEARNING TECHNIQUES

Pushpendra Dwivedi, C. S. Raghuvanshi , Hari Om Sharan
Faculty of Engineering & Technology, Rama University, Kanpur 209217, Uttar Pradesh, INDIA
E-mail: pushpendradwivedi10@gmail.com

Abstract: ***Malware remains an enduring and evolving threat in the digital landscape, necessitating innovative approaches for its detection and classification. The study underscores the significance of feature fusion, amalgamating diverse attributes from various sources to encapsulate both static and dynamic facets of malware. Traditional single-feature methods exhibit limitations in precision, motivating the exploration of multiple characteristics for fusion and employing a unified learning algorithm for classifying malware families. The research methodology involves meticulous feature extraction, followed by the utilization of KNN, XGBoost, DecisionTree, and random forest algorithms for classification, utilizing the most critical features. Experimental results underscore the significant improvement in classification accuracy compared to conventional methods, effectively reducing false positives fusion improves malware classification accuracy by 99.11% using dynamic features, 97.31% using static features, and 99.88% using a hybrid analysis compared to the conventional method. Moreover, the study focuses on merging Convolutional Neural Network (CNN) deep learning models with feature fusion specifically for Portable Executables (PE) files, achieving a remarkable accuracy of 99.18% in discerning between benign and malicious software. This synthesis of deep learning and feature fusion remarkably fortifies mal-ware classification efficacy, offering a potent solution to combat evolving cyber threats.***

Keywords: ***deep learning, malware classification, feature fusion, machine learning, PE files***

1. INTRODUCTION

Effective malware classification and detection has become a crucial issue in cybersecurity due to the alarming increase of malware threats in the digital domain. Cybercriminals are always finding new ways to trick computer systems, networks, and the data stored within them with their malicious software. Malware analysis has been a subject of constant innovation in response to this concern. One of the most common ways to monitor software activity is by looking at the system API call sequence. This is because it logs every single thing the application does, such accessing files or networks. Every API call in the sequence relies on the name of the API and its arguments [1]. Always specified as name=value pairs, the arguments of an API request can have any number from zero to many. An assortment of feature engineering techniques is offered for processing data pertaining to behavior. Assuming the API name is a string, we may get the N most prevalent n-gram features (where n = 1, 2) of it. Parameters might be of many different types, including texts, integers, addresses, and more, making feature extraction a challenging task. Static and dynamic examination of malware are the two primary methods that may be utilized in order to get information about the characteristics of malware. The utilization of static characteristics al-lows for the extraction of significant information concerning the compositional particulars of the file. For the purpose of static malware analysis, PE- section, import, header, byte, and Opcode histograms are frequently utilized [2].

However, with these characteristics, it is possible that vital data concerning advanced malware techniques such as obfuscation, metamorphism, mutation, and oligomorphic code that are employed to prevent recognition is omitted. The actions and behaviors of the executable file may be captured by dynamic malware analysis, which can then be used for the recognition and categorization of malware [3]. When it comes to dynamic malware analysis, the feature set that is utilized the most frequently is the API call sequencing. This is due to the fact that it not only records the communication of the binary with the different system instances, but additionally reveals the motive behind the construction of the virus. In addition, researchers made use of a technique known as hybrid analysis, which employed an accumulation of static and dynamic characteristics in order accurately identify vulnerabilities and increase efficiency [4].

Because of its capacity to learn sophisticated representations and patterns from complicated data, deep learning has attracted a lot of interest and demonstrated potential in malware identification and classification. Deep learning models, like CNNs and RNNs, may automatically learn hierarchical representations from raw data (such opcode sequences, byte-level information, or binary code) without using constructed features [5]. Capturing complex virus patterns has been made possible by this capacity to automatically learn characteristics from data. By inspecting sequences of system calls, API calls, or network traffic, deep learning models may evaluate software behavioral patterns. Malware analysis and classification using behavioral sequences has made use of recurrent neural networks, particularly Long Short-Term Memory (LSTM) networks. For thorough malware detection, deep learning approaches can combine static (file-based) and dynamic (behavior-based) analysis. Enhanced classification accuracy may be achieved by combining characteristics retrieved from static and dynamic analyses [6]. Static analysis features include file headers and byte sequences, while dynamic analysis features include API calls and system actions. The use of deep learning algorithms to identify

malware and counter its evasion tactics has been investigated by re-searchers. Making models that are less prone to escape has been achieved via the use of adversarial training and robustness strategies. Malware analysis has made use of ensemble approaches that incorporate deep learning architectures, domain adaptability, and transfer learning techniques (pre-trained models). In situations when there is a lack of labeled data, these methods improve classification accuracy by drawing on information from big datasets or adjacent fields [7].

Feature fusion approaches integrated into machine learning frameworks are an interesting and potentially fruitful direction for malware classification to go [8]. This method takes into account the fact that malware is complex and displays a wide range of behaviors and traits. Feature fusion acknowledges that a holistic view combining many attributes taken from different sources, such as file-based features, network traffic patterns, and behavioral traits, is necessary for full malware classification. The combi-nation of these varied characteristics enhances the accuracy and resilience of classification models by giving a fuller picture of harmful software [9]. The use of feature fusion in conjunction with machine learning has recently demonstrated encouraging results in improving malware classification accuracy. These methods are resistant to malware evasion tactics and increase detection rates simultaneously. Contributing to the continuous efforts to strengthen cybersecurity and defend digital environments from emerging threats, this research explores the integration of state-of-the-art machine learning models with feature fusion methodologies:

In this work, we look at the most recent developments and trends in malware categorization using feature fusion and machine learning together. What follows is an explanation of malware categorization, feature fusion, and machine learning. In this paper, we present experimental data and investigate how these methods address modern cybersecurity challenges. This study will be valuable for researchers, cybersecurity experts, and companies looking for strategies and solutions to combat emerging malware threats.

2. RELATED WORKS

Analysis of malware samples is performed to identify the properties that may be applied to determine them. Since malware is becoming more sophisticated in the lifecycle, knowledge about cryptic malware protection has emerged as a critical issue in malware detection, according to machine learning methodologies [10]. Additionally, there still are two types of malware analysis that are often used in the process of identifying malicious applications [11]. malware detection techniques based on ML strategies uses feature extraction for the analysis. These features (API calls, Assembly, and Binary) [12] used machine learning methodologies for classifying malware.

Many different approaches to identifying and classifying malware have been developed in the field of malware classification. When faced with novel, unanticipated dangers, traditional methods that depend on patterns, such signature-based detection, frequently fail [13]. There is a chance of false alarms when using heuristic-based algorithms to detect possible dangers based on patterns of behavior. Machine learning has brought about a dramatic change by using supervised, unsupervised, and deep learning algorithms to analyze large collections of malware data and distinguish between safe and dangerous software. Furthermore, hazardous acts can be detected through the use of behavioral analysis, which is carried out in controlled situations [14]. For better ac-curacy, some sophisticated strategies merge signatures, heuristics, behavioral analysis, and machine learning into hybrid models that incorporate numerous detection methods [15]. To learn how malware affects systems, dynamic analysis is used, which comprises seeing it in action in a controlled environment in real-time. In addition, a powerful method that provides a comprehensive view of malware activity is the fusion of distinct data derived from different sources [16]. Ongoing research aims to improve and adapt these approaches to tackle the ever-changing cyber threats as the cybersecurity land-scape changes.

2.1. SIGNATURE BASED MALWARE DETECTION APPROACH

conventional malware that was wide and accessible, modern malware is more specialized, stealth and has a long-term presence compared to conventional malware that was only executed once [17]. The identification of zero-day infection is difficult since it utilizes newer vulnerabilities that have not yet been disclosed [18]. A wide range of computer science fields now use Artificial Intelligence, ML, and deep learning methodologies, from NLP to malware detection strategies. Author [19] has researched Android malware; a multi-feature consensus-based decision fusion adaptive identification component is now being created to utilize this malware (MCDF). To classify malware samples, Srndic et al. [20] employed static analysis in conjunction with ma-chine learning techniques. Two separate file types were investigated in this research. Malware authors increasingly embed resource-depleting executables in PDF and SWF files. This study analyzed 440,000 PDFs and 40,000 SWFs. This technology's architecture made it possible to identify malicious code in Adobe PDF and Flash (SWF) files.

An anti-virus, or malware detection system relies heavily on the use of signatures to identify suspicious activity. This approach works by searching a vast dataset of sig-natures for specific patterns of viruses. The signature-based method searches for disruptions by referring to a previously specified list of known attacks. Regardless of the fact that this configuration is capable of identifying malware in a wide variety of ap-plications, it needs the regular updating of the specified signature database to maintain its effectiveness. As a result, it is less successful in detecting harmful workouts when using the signature-based method, owing to the constantly evolving nature of versatile malware [21]. Metaheuristic approaches are adopted by the anti-virus provider which can effectively identify the malicious codes to manage their signature [22].

Feature extraction tools like PeView, PeExplorer, PsStudio,

Hash Generator are for static feature extraction. Static analysis at code level is achieved using disassembler tools for example Lida, Cpstone, IDA Pro. Malware static features like N-gram [23](n-gram 3: 'mail', 'ili, 'ftw'), String [23]('APIcallname', 'mytime', 'kernal32'), Opcode [15]('ADD', 'SUB', 'MOV, 'PUSH'), Hash Values ('e5dadf6524624f79c3127e247f04b548'), PE Header information [24]('field value', 'checksum', 'size', 'symbol') are extracted for analysis . The challenge of signature-based identification may be reduced to a simple one of string matching. Basically, this implies that it continues looking for a pattern or substring in a huge string dataset. Almost all of the computing time is spent to just this procedure (45 percent to 75 percent of the time) [25]. Aho-Corasick and Boyer-Moore are two of the most often used algorithms for string matching. De-obfuscating every piece of malware is quite difficult, despite the fact that many unpacking techniques are pre-sent.

WU Bin et al. (2015) [26] proposed a malware detection model for the mobile phone based on artificial immune based system. As well as varying detectors, a clone and mutation method is applied to increase the detection accuracy. Token-based re-semblance and character-based resemblance were combined to create a new similarity matrix, and it was also shown that existing characteristics are specific examples of fuzzy token similarity. Jiannan Wang et al. (2011) [27] developed a signature-based system to solve the problem of fuzzy-token similarity joins. In comparison to other existing sig-nature techniques, it is found that the token-sensitive approach is better. Edit similarity was included as an extension to current signature systems for edit distance. The study in [8] suggested ScaleMalNet, a deep learning system for detecting zero-day malware that uses image processing, dynamic analysis, and static data. To define malware, [28] pro-posed a method based on features of behavior. To get elimination of the proposed model, they gather API call traces from malware samples in a controlled virtual environment and run dynamic inspection on a dataset of usually early malware. In order to create more advanced characteristics, or "actions," the traces are first processed. Ac-cording to the methods proposed by Arivudainambi, Varun, et al. [29], malicious traffic may be identified by network traffic analysis. Using PCA was a must for better anti-network traffic methodological techniques. The proposed method was tested in various sandboxes, including Noriben, Cuckoo, and Limon, by running 1,000 malicious files. The method's success rate in identifying malware was 99 percent.

The usage of signatures to detect unusual behavior is crucial to anti-virus or mal-ware detection systems. In order for this method to detect viruses, it searches a large database of signatures for certain patterns. A previously defined list of known assaults is used by the signature-based technique to seek for disruptions. The setup may detect malware in many different contexts, but it requires the provided signature database to be updated often for it to continue working. Since flexible malware is always changing, the signature-based technique has limited efficacy in detecting malicious exercises [21]. The anti-virus supplier uses metaheuristic methodologies to maintain signatures and successfully identify harmful software [22].

The static feature extraction programs such as PeView, PeExplorer, PsStudio, and HashGenerator. Lida, Cpstone, and IDA Pro are disassembler tools that may be used to do static analysis at the code level. In order to analyze the malware, certain static features are extracted, such as N-grams, strings, opcodes, hash values, and PE header in-formation. If string matching proves to be too difficult, signature-based identification may become as easy as pie. What this means in practice is that it searches through a massive string collection in search of a pattern or substring. This one process accounts for nearly all of the processing time (between 45 and 75 percent) [25]. Popular string-matching algorithms include Aho-Corasick and Boyer-Moore. Despite the availability of several unpacking approaches, decrypting every piece of malware re-mains a formidable challenge. Utilizing supervised machine learning techniques, Narayanan et al. (2016) [30] built a malware classifier that was able to handle polymorphic.

2.2. BEHAVIOUR BASED MALWARE DETECTION APPROACH

Anomaly refers to a malfunction caused by malicious files and is taught into the behavior-based approaches in two ways. Malicious files are those that display anomalous behavior that is inconsistent with the stored behavior of normal files.

Behavioral-based malware detections approaches are discussed in detail in this section. The advanced methods are brought up to identify malware, Bailey et al. (2007) [31] suggested a method that recorded malware's API calls. A new hybrid method, HDM-Analyzer, was proposed by Eskandari et al. (2013) [32], taking into account both dynamic and static inquiry points of interest, while keeping precision at a reasonable level. Because of this, HDM-Analyzer can forecast that most of the fundamental leadership is based on real data, and so has little performance degradation. Sheen et al. (2015) [19] developed MCDF. Malicious record characteristics like the consent-based features and API call-based features are evaluated in order to provide a better discovery by merging the classifiers' choices using the collective method based on the probability hypothesis, which is used to construct a group of classifiers.

Table.1. Tools used for static and dynamic analysis

Static Analysis Tools	Dynamic Analysis Tools
IDA Pro (dissembler)	ProcMon (logs lve system activity)
Ghidra (dissembler)	PeStudio (Windows executable analyzer)
PeView (PE header information)	Process Hacker (Gathering information of process)
UPX	Wireshark (packet analysis tool)
YARA (string matching)	TCPdump (TCP/IP packet analyser)
x64dbg (reverse engineering)	Regshot (snapshot of registry related files)
HxD	VmWare/VitualBox (virtual machine)
PE-bear	Comodo IMA (malware analysis sandbox)

PeStudio (analyzing executables)	Cuckoo Sandbox
IOCFinder	RegMon (registry monitoring)
Static Analysis Tools	Dynamic Analysis Tools
IDA Pro (dissembler)	ProcMon (logs lve system activity)

Utilizing supervised machine learning techniques, Narayanan et al. (2016) [30] built a malware classifier that was able to handle polymorphic. Ming et al. (2017) [6] have developed a substitution attack that affects behavior-based requirements to cover similar behaviors. The main attack approach is to replace a graph of system call de-pendency with its semantically equivalent variants so that the comparable malware test's secret unique family becomes characteristically distinctive. Malware researchers should thus devote more time and effort to the re-examination of identical samples that may have recently been studied, as a result of this.

Deep learning is just one method within the larger field of machine learning [33]. It can be trained with data that is neither organized nor tagged. It collects data, processes it, and then forms conclusions based on patterns it finds about itself; this is quite similar to how the human brain works. Deep learning relies on neurons as its foundation [8].

3. PROPOSED APPROACH

For accurate malware detection, using the relevant algorithm is important. When estimating supervised learning models based on feature engineering, the prior top performer is Support Vector Machines (SVM) which is used [34]. After a thorough study, the CNN model was selected because it performed exceptionally well with many feature sets obtained from different sources, including static analysis, dynamic analysis, and binary-to-image conversion techniques. To demonstrate how effective feature fusion is in enhancing classification accuracy, the chosen CNN model is applied to the evaluation of the fused combination dataset for malware classification. Figure 1 shows the propose approach of malware classification The effectiveness of feature fusion in improving malware classification accuracy is demonstrated in this all-encompassing method, which uses DL models, optimizes hyperparameters, evaluates performance across different feature sets, and finally uses a selected model (CNN) for both the fused feature set and individual feature sets.

Using information such as sections, imports, APIs, and pictures retrieved from Portable Executable (PE) files, we are interested in developing a method that uses a convolutional neural network (CNN) feature fusion approach to detect and categorize malware instances. The approach details the steps to build a malware detection classifier using convolutional neural networks (CNNs). The procedure begins with preprocessing and continues with numerous rounds of CNN layers (convolutional, pooling, dense, and dropout) to train and categorize malware samples using the fused characteristics retrieved from various parts of PE files. In order to identify and categorize malware samples, this study technique lays forth a systematic way to use a convolutional neural network (CNN) feature fusion model using various properties extracted from PE files. Everything from extracting and fusing features to developing a convolutional neural network (CNN) model for malware classification is part of it.

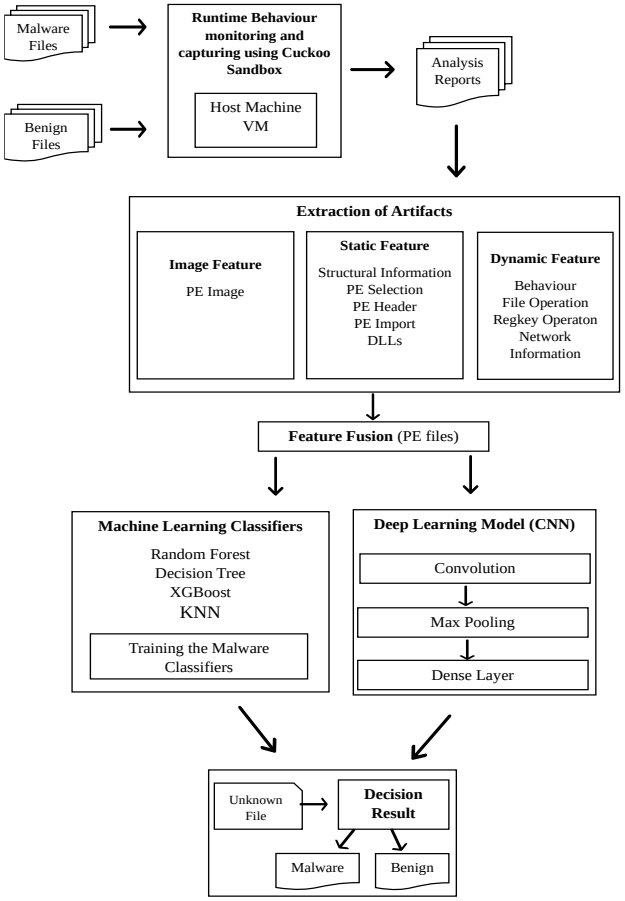

Fig.1. Proposed classification scheme

3.1. DATA SET

The dataset is structured to assist in the classification and analysis of different malware types based on their families. It comprises a diverse set of malware types, each belonging to specific families, and includes the number of samples available for each malware family. The dataset covers a total of 29710 samples across various malware types and their associated families. It comprises a wide range of malicious software, including viruses (Krepper.30760), worms (Yuner.A), backdoors (Agent, 1024 samples), trojan downloaders (Tugaspay.A, 3652 samples), and trojan removers (Renos, 1880 samples), among other types of malwares. It also includes samples from rogue malware, trojans, virtools, and trojan dropper families, with different numbers of samples from each family adding diversity to the collection. Datasets like this one are crucial to cybersecurity researchers because they allow for the testing and refinement of machine learning models that can distinguish between and categorize malware families. Nevertheless, it is crucial to address class imbalance appropriately during model training and assessment to guarantee accurate and robust classification results, since the dataset's potential might be affected by an uneven distribution

of samples across different malware families**.**

3.2. PROPOSED ALGORITHM

Algorithm. Malware Classification by Integrating Feature Fusion with machine and Deep Learning

Input. PE_section: {ps1, ps2, ..., psm}, PE_import: {pi1, pi2, ..., pin}, PE_API: {pa1, pa2, ..., pap}, PE_image: {pim1, pim2, ..., pimq}

Output. Output predictions O1 representing the probability of the sample being classified as Malware or Benign

Feature Integration. Combine PE features (sections, imports, APIs, images) to create fusion feature = {F1, F2, ..., Ft}, where m + n + p + q = t.

Preprocessing. Preprocess fusion_feature to obtain pre-processed feature set: {FS1, FS2, ..., FSt}.

CNN Operations.

for each epoch (e) from 1 to e do

for each dataset sample (d) from 1 to d do

- Perform 1D Convolution with kernel filter (k) and filter length (l) t to obtain intermediate convolutional features.

- Apply Max Pooling with pool size (b) on the convolutional features to extract essential features

- Flatten pooled features to get a flattened representation

- Use Dense layers (Dense) with units (x) to learn meaningful representations

- Apply Dropout (Dropout) to prevent overfitting

- Employ Sigmoid activation (Sigmoid) to generate output predictions O1, indicating malware or benign probability.

end for

end for

4. EXPERIMENTAL RESULTS

A multi-view feature fusion approach for effective malware classification using Deep Learning refers to a method that combines multiple perspectives or representations of malware samples to improve their classification accuracy. Traditionally, malware classification has relied on individual feature sets or representations, such as static features extracted from the binary file, dynamic features obtained from monitoring its execution, or behavior-based features derived from analyzing its actions. However, each feature set may have limitations in capturing the full complexity of malware, leading to suboptimal classification results.

By combining complementary information from multiple views, the multi-view feature fusion approach aims to enhance the classification accuracy of malware samples. It leverages the power of Deep Learning to automatically learn meaningful representations from diverse feature sets, leading to more robust and effective malware classification systems. The experiment conducted with the top 10, 20, 30, 40 and 50 features using Random_forest, XGBoost, Decision Tree, KNN algorithms. The results show that Random Forest using top 40 features achieved highest accuracy of 97.31%. The experiment conducted with the top 10, 20, 30, 40, 50, 60 and 70 features using Random_forest, XGBoost, Decision Tree, KNN algorithms.

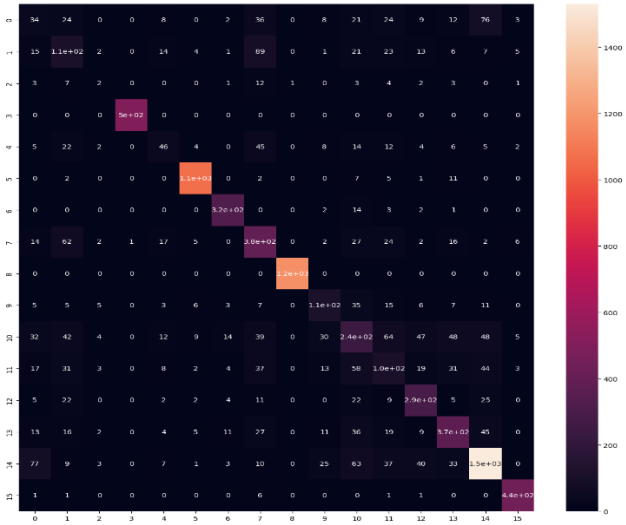

Fig.2. Confusion matrix for rootkit data set

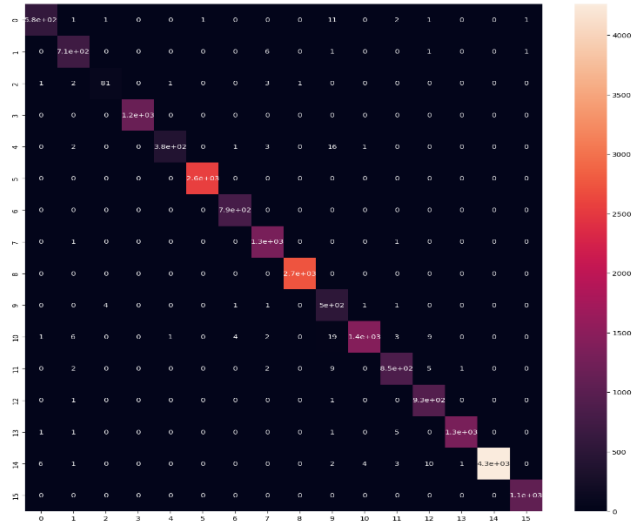

Fig.3. Confusion matric for backdoor dataset

The discussion begins by detailing the datasets used for conducting experiments and analyses in the field of malware classification. The dataset consisted of eight distinct malware types spanning across 16 malware families, comprising a total of 29,710 samples. It then delves into the different techniques employed for analysis. Firstly, the Convolutional Neural Network (CNN) was utilized with 100 epochs, showcasing im-pressive training results with a loss of 0.0398 and an accuracy of 0.9868 concerning the sequential model. Figure 2 and figure 3 shows the confusion matrix generated for the rootkit and backdoor dataset respectively. Testing with the sequential model, encompassing a larger dataset of 711,342 samples, yielded a slightly lower accuracy of 75.17%. Table 2 and 3 shows the number of features set selected and the accuracy of the result obtained by the various algorithm (RF, XGB, DT and KNN). The analysis was further

segregated into Static Analysis, Dynamic Analysis, and Hybrid Analysis. In Static Analysis, various feature subsets (top 10, 20, 30, 40, 50) were tested using Random Forest, XGBoost, Decision Tree, and KNN algorithms. Notably, employing the top 40 features with Random Forest exhibited the highest accuracy of 97.31%.

Conversely, Dynamic Analysis involved testing different feature subsets (top 10, 20, 30, 40, 50, 60, 70) with the same algorithms, revealing that utilizing the top 40 features with Random Forest led to the highest accuracy of 99.11%. The Hybrid Analysis, combining 40 static and 60 dynamic features, demonstrated exceptional accuracy rates: 99.65% for Random Forest, 99.89% for XGBoost, 99.10% for Decision Tree, and 93.84% for KNN algorithms.

Table.2. Static Analysis with feature numbers and algorithms

Feature_no.	Random_forest	XGBoost	DecisionTree	KNN
10	96.54337296	95.75077059	95.50858653	93.52708058
20	96.85160722	96.12505504	95.97093791	93.81329811
30	97.22589168	96.67547336	96.01497138	93.79128137
40	97.31395861	97.09379128	96.23513871	93.83531484
50	97.18185821	97.00572435	96.3672391	93.81329811

Comparatively, the multi-feature approach proved superior to using a single feature, showcasing the efficacy of feature fusion methodology. Specifically, it achieved an accuracy of 97.31% for static analysis, 99.11% for dynamic analysis, and an impressive 99.64% for hybrid analysis. While highlighting these achievements, it's crucial to acknowledge both the ad-vantages and limitations of these methodologies in malware classification. The multi-feature approach demonstrates substantial improvements in accuracy, but the field may still face challenges in certain scenarios, such as evasion techniques employed by malicious entities.

Table.3. Dynamic Analysis with feature numbers and algorithms

Feature_no.	Random_forest	XGBoost	DecisionTree	KNN
10	96.52135623	95.97093791	96.27917217	94.25363276
20	98.59092911	98.28269485	97.86437693	95.83883752
30	98.70101277	98.94319683	98.23866138	90.22457067
40	99.09731396	99.07529723	98.45882871	90.81902246
50	99.03126376	99.03126376	98.43681198	90.40070454
60	98.83311317	99.11933069	98.56891237	90.29062087
70	99.00924703	99.07529723	98.39277851	90.29062087

Nonetheless, this comprehensive analysis showcases the potential and effectiveness of utilizing diverse features and models for enhanced malware classification.

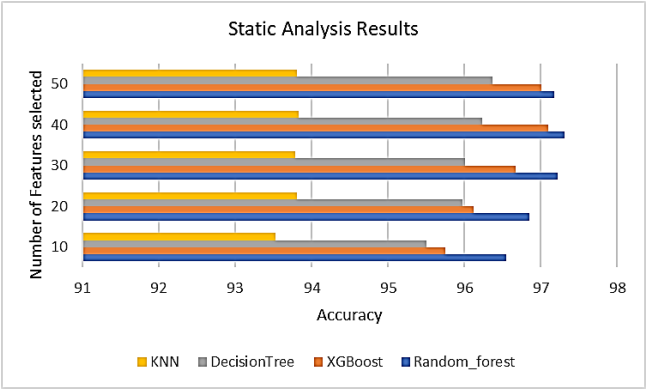

Fig.4. Accuracy in static analysis with the number of features set selected

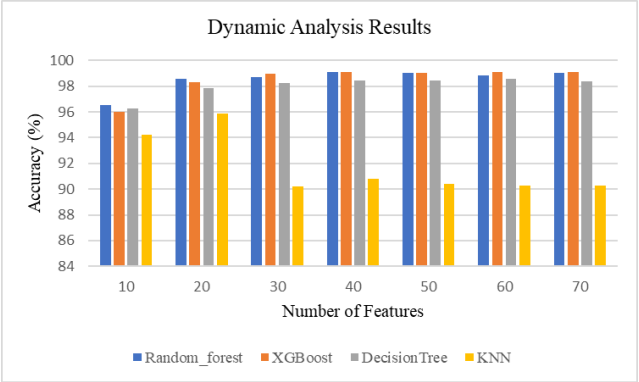

Fig.5. Accuracy in dynamic analysis with the number of features set selected

The results as shown in figure 4 Random Forest using top 40 features and accuracy 97.31% and figure 5 depicts that XGBoost using top 60 features achieved highest accuracy of 99.11% for the dynamic analysis and for hybrid analysis top 40 static and 60 dynamic features were selected. An accuracy of 99.64773227653016, 99.88991633641568, 99.09731395860855, 93.83531483927786 was observed for the Random_forest, XGBoost, DecisionTree, KNN algorithms. In comparison to using a single feature, the proposed multi-feature approach for malware classification provides better results. Thus, the feature fusion methodology achieved an accuracy of 97.31% for static analysis and 99.11% for dynamic analysis. Also, hybrid analysis achieves 99.64%. Number of Epoch used are 100 and training results shows a loss of 0.0398 and an accuracy of 0.9868 while considering the sequential_1 model. precision recall f1-score support 99.18% accuracy. Testing results with epoch 100 using sequential_4 model with a total 711,342 shows an accuracy of 75.17%.

5. CONCLUSION

Finally, the multi-view feature fusion method has great potential to improve malware classification systems' accuracy. Malware classification methods that depend on static, dynamic, or behavior-based elements alone have historically failed to capture the full complexity of malware, leading to less-than-ideal results. This method takes use of Deep Learning's capabilities to automatically build meaningful representations from different feature sets by combining separate but complementary data sets from different sources. Notable

accomplishments were displayed by experimental findings that utilized the Random Forest, XGBoost, Decision Tree, and KNN algorithms. Use of the top 40 characteristics yielded a 99.11% accuracy rate for dynamic analysis and a 97.31% rate for static analysis; a 99.64% accuracy rate was attained via a hybrid analysis that combined 40 static features with 60 dynamic features. By demonstrating significant gains in classification accuracy across different studies, these results demonstrate that the feature fusion approach is more effective than employing individual features. Malicious actors may alter binary files to avoid detection, thus it's important to use Deep Learning or Machine Learning models that can overcome evasion techniques to make malware detection systems more resilient. Combating developing threats and improving the overall efficiency of malware categorization approaches requires more studies and breakthroughs in model building.

Several important aspects will be improved in the future of malware categorization. Among them, we may find ways to improve the interpretability of models, investigate more complex deep learning architectures, strengthen machine learning models against adversarial assaults, and enhance feature engineering to provide more in-formative and robust feature sets. For effective processing of big datasets and real-time threat identification, scalability, and advancements in unsupervised learning approaches are crucial. For complete malware identification, it is crucial to prioritize user privacy, enhance behavioral analysis, and promote collaborative defensive systems. To keep things moving forward and make sure future malware classification algorithms are strong, it is essential to constantly generate and maintain datasets. If these areas are addressed, the field's capacity to fight changing cyber threats will be greatly enhanced.

ACKNOWLEDGEMENT

I would like to express our sincere gratitude to Dr. C. S. Raghuvanshi and Dr. Hari Om Sharan for their unwavering support and invaluable contributions to this research study. Their equal dedication and guidance in all aspects of the research have been instrumental in shaping the course of our work.

I also extend our thanks to all the authors, including Dr. C. S. Raghuvanshi and Dr. Hari Om Sharan, for their collective efforts in thoroughly reviewing and providing valuable insights that have significantly enriched the content of this manuscript. The collaborative spirit and unanimous agreement on the final version of the paper underscore the strength of our teamwork. Their expertise, mentorship, and commitment have been pivotal in the successful completion of this research, and we are truly grateful for their enduring support.

REFERENCES

[1] A A. P. Namanya, A. Cullen, I. U. Awan, and J. P. Disso, “The World of Malware: An Overview,” Proceedings - 2018 IEEE 6th International Conference on Future Internet of Things and Cloud, FiCloud 2018, pp. 420–427, 2018, doi: 10.1109/FiCloud.2018.00067.

[2] A. M. Abiola and M. F. Marhusin, “Signature-based malware detection using sequences of N-grams,” International Journal of Engineering and Technology(UAE), vol. 7, no. 4, pp. 120–125, 2018, doi: 10.14419/ijet.v7i4.15.21432.

[3] A. Namavar Jahromi et al., “An improved two-hidden-layer extreme learning machine for malware hunting,” Comput Secur, vol. 89, p. 101655, 2020, doi: 10.1016/j.cose.2019.101655.

[4] M. Rabbani, Y. L. Wang, R. Khoshkangini, H. Jelodar, R. Zhao, and P. Hu, “A hybrid machine learning approach for malicious behaviour detection and recognition in cloud computing,” Journal of Network and Computer Applications, vol. 151, p. 102507, 2020, doi: 10.1016/j.jnca.2019.102507.

[5] Q. Le, O. Boydell, B. Mac Namee, and M. Scanlon, “Deep learning at the shallow end: Malware classification for non-domain experts,” Proceedings of the Digital Forensic Research Conference, DFRWS 2018 USA, pp. S118–S126, 2018, doi: 10.1016/j.diin.2018.04.024.

[6] J. Ming, Z. Xin, P. Lan, D. Wu, P. Liu, and B. Mao, “Impeding behavior-based malware analysis via replacement attacks to malware specifications,” Journal of Computer Virology and Hacking Techniques, vol. 13, no. 3, pp. 193–207, 2017, doi: 10.1007/s11416-016-0281-3.

[7] A. Boukhtouta, S. A. Mokhov, N. E. Lakhdari, M. Debbabi, and J. Paquet, “Network malware classification comparison using DPI and flow packet headers,” Journal of Computer Virology and Hacking Techniques, vol. 12, no. 2, pp. 69–100, 2016, doi: 10.1007/s11416-015-0247-x.

[8] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, and S. Venkatraman, “Robust Intelligent Malware De-tection Using Deep Learning,” IEEE Access, vol. 7, pp. 46717–46738, 2019, doi: 10.1109/ACCESS.2019.2906934.

[9] Y. Ding, J. Hu, W. Xu, and X. Zhang, “A DEEP FEATURE FUSION METHOD FOR ANDROID MALWARE DETEC-TION,” 2019 International Conference on Machine Learning and Cybernetics (ICMLC), pp. 1–6.

[10] A. Souri and R. Hosseini, “A state-of-the-art survey of malware detection approaches using data mining techniques,” Human-centric Computing and Information Sciences, vol. 8, no. 1. 2018. doi: 10.1186/s13673-018-0125-x.

[11] M. Ijaz, M. H. Durad, and M. Ismail, “Static and Dynamic Malware Analysis Using Machine Learning,” Proceedings of 2019 16th International Bhurban Conference on Applied Sciences and Technology, IBCAST 2019, pp. 687–691, 2019, doi: 10.1109/IBCAST.2019.8667136.

[12] J. Singh and J. Singh, “Assessment of supervised machine learning algorithms using dynamic API calls for malware detection,” International Journal of Computers and Applications, vol. 44, no. 3, pp. 270–277, 2022, doi: 10.1080/1206212X.2020.1732641.

[13] A. Abusitta, M. Q. Li, and B. C. M. Fung, “Journal of Information Security and Applications Malware classification and composition analysis : A survey of recent developments,” Journal of Information Security and Applications, vol. 59, no. April, p. 102828, 2021, doi: 10.1016/j.jisa.2021.102828.

[14] S. Dambra, A. Vitale, J. Caballero, and D. Balzarotti, "Decoding the Secrets of Machine Learning in Windows Malware Classification : A Deep Dive into Datasets , Features , and Model Performance".

[15] J. Sexton, C. Storlie, and B. Anderson, "Subroutine based detection of APT malware," Journal of Computer Virology and Hacking Techniques, vol. 12, no. 4, pp. 225–233, 2016, doi: 10.1007/s11416-015-0258-7.

[16] P. Dwivedi and H. Sharan, "Analysis and Detection of Evolutionary Malware: A Review," Int J Comput Appl, vol. 174, no. 20, pp. 42–45, 2021, doi: 10.5120/ijca2021921005.

[17] E. Gandotra, D. Bansal, and S. Sofat, "Malware Analysis and Classification: A Survey," Journal of Information Security, vol. 05, no. 02, pp. 56–64, 2014, doi: 10.4236/jis.2014.52006.

[18] R. Kaur and M. Singh, "Hybrid Real-time Zero-day Malware Analysis and Reporting System," International Journal of Information Technology and Computer Science, vol. 8, no. 4, pp. 63–73, 2016, doi: 10.5815/ijitcs.2016.04.08.

[19] S. Sheen, R. Anitha, and V. Natarajan, "Android based malware detection using a multifeature collaborative decision fusion approach," Neurocomputing, vol. 151, no. P2, pp. 905–912, 2015, doi: 10.1016/j.neucom.2014.10.004.

[20] N. Šrndić and P. Laskov, "Hidost: a static machine-learning-based detector of malicious files," EURASIP J Inf Secur, vol. 2016, no. 1, pp. 1–20, 2016, doi: 10.1186/s13635-016-0045-0.

[21] M. Al-Asli and T. A. Ghaleb, "Review of signature-based techniques in antivirus products," 2019 International Con-ference on Computer and Information Sciences, ICCIS 2019, pp. 1–6, 2019, doi: 10.1109/ICCISci.2019.8716381.

[22] S. Sibi Chakkaravarthy, D. Sangeetha, and V. Vaidehi, "A Survey on malware analysis and mitigation techniques," Comput Sci Rev, vol. 32, pp. 1–23, 2019, doi: 10.1016/j.cosrev.2019.01.002.

[23] H. Zhang, X. Xiao, F. Mercaldo, S. Ni, F. Martinelli, and A. K. Sangaiah, "Classification of ransomware families with machine learning based on N-gram of opcodes," Future Generation Computer Systems, vol. 90, pp. 211–221, 2019, doi: 10.1016/j.future.2018.07.052.

[24] features and public APT reports," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 10332 LNCS, pp. 288–305, 2017, doi: 10.1007/978-3-319-60080-2_21.

[25] G. Laurenza, L. Aniello, R. Lazzeretti, and R. Baldoni, "Malware triage based on static.

[26] Y.-H. Choi, M.-Y. Jung, and S.-W. Seo, "L+1-MWM: A Fast Pattern Matching Algorithm for High-Speed Packet Fil-tering," pp. 2288–2296, 2008, doi: 10.1109/infocom.2008.297.

[27] B. Wu, X. Lin, W. D. Li, T. L. Lu, and D. M. Zhang, "Smartphone malware detection model based on artificial immune system in cloud computing," Beijing Youdian Daxue Xuebao/Journal of Beijing University of Posts and Telecommu-nications, vol. 38, no. 4, pp. 33–37, 2015, doi: 10.13190/j.jbupt.2015.04.008.

[28] J. Wang, G. Li, and J. Fe, "Fast-join: An efficient method for fuzzy token matching based string similarity join," Proc Int Conf Data Eng, pp. 458–469, 2011, doi: 10.1109/ICDE.2011.5767865.

[29] H. S. Galal, Y. B. Mahdy, and M. A. Atiea, "Behavior-based features model for malware detection," Journal of Computer Virology and Hacking Techniques, vol. 12, no. 2, pp. 59–67, 2016, doi: 10.1007/s11416-015-0244-0.

[30] D. Arivudainambi, V. K. Varun, S. C. S., and P. Visu, "Malware traffic classification using principal component analysis and artificial neural network for extreme surveillance," Comput Commun, vol. 147, no. July, pp. 50–57, 2019, doi: 10.1016/j.comcom.2019.08.003.

[31] B. N. Narayanan, O. Djaneye-Boundjou, and T. M. Kebede, "Performance analysis of machine learning and pattern recognition algorithms for Malware classification," Proceedings of the IEEE National Aerospace Electronics Confer-ence, NAECON, vol. 0, pp. 338–342, 2016, doi: 10.1109/NAECON.2016.7856826.

[32] M. Bailey, J. Oberheide, J. Andersen, Z. M. Mao, F. Jahanian, and J. Nazario, "Automated classification and analysis of Internet malware," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 4637 LNCS, pp. 178–197, 2007, doi: 10.1007/978-3-540-74320-0_10.

[33] M. Eskandari, Z. Khorshidpour, and S. Hashemi, "HDM-Analyser: A hybrid analysis approach based on data mining techniques for malware detection," Journal in Computer Virology, vol. 9, no. 2, pp. 77–93, 2013, doi: 10.1007/s11416-013-0181-8.

[34] J. Hemalatha, S. A. Roseline, S. Geetha, S. Kadry, and R. Damaševičius, "An efficient densenet-based deep learning model for Malware detection," Entropy, vol. 23, no. 3, pp. 1–23, 2021, doi: 10.3390/e23030344.

[35] J. Sun, K. Yan, X. Liu, C. Yang, and Y. Fu, "Malware detection on android smartphones using keywords vector and SVM," Proceedings - 16th IEEE/ACIS International Conference on Computer and Information Science, ICIS 2017, no. 61502134, pp. 833–838, 2017, doi: 10.1109/ICIS.2017.7960108.