MALWARE CLASSIFICATION WITH MACHINE LEARNING USING MULTI VIEW FEATURE SELECTION AND FUSION APPROACH

Pushpendra Dwivedi, C. S. Raghuvanshi and Hari Om Sharan

Department of Computer Science & Engineering, Rama University, Kanpur 209217, Uttar Pradesh, INDIA

E-mail: pushpendradwivedi10@gmail.com

Abstract: *Malware continues to pose a persistent and dynamic threat in the digital realm, demanding innovative detection and classification strategies. This study emphasizes the importance of feature fusion, which integrates diverse attributes from multiple sources to capture both static and dynamic aspects of malware comprehensively. Conventional single-feature approaches exhibit precision limitations, driving the exploration of multiple characteristics for fusion and the adoption of a unified learning algorithm for malware family classification. The research methodology involves meticulous feature extraction, followed by the application of KNN, XGBoost, DecisionTree, and random forest algorithms for classification, leveraging the most critical features. Experimental findings demonstrate a significant enhancement in classification accuracy compared to traditional methods, effectively reducing false positives. Fusion techniques enhance malware classification accuracy by 99.11% with dynamic features, 97.31% with static features, and 99.88% with hybrid analysis, surpassing conventional methodologies***.

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

1. INTRODUCTION

A major concern in cybersecurity nowadays is effective malware categorization and detection because of the worrying rise in malware threats in the digital sphere. Cybercriminals are always coming up with new strategies to use their malicious software to deceive computer systems, networks, and the data that is contained therein. In response to this worry, malware analysis has been the topic of ongoing innovation. Examining the sequence of system API calls is one of the most popular methods for keeping an eye on programme activities. This is so that everything the application performs, including file and network access, is recorded. The names of the API and its parameters are necessary for each API call in the series [1]. An API request's parameters, which are always given as name=value pairs, can contain any number between 0 and many. A variety of feature engineering methods are available for handling behavior-related data processing. We may obtain the N most common n-gram characteristics (where n = 1, 2) of the API name, assuming it is a string. Feature extraction is a difficult operation as parameters might be many various types, such as texts, numbers, locations, and more. The two main techniques that may be applied to learn more about the features of malware are static and dynamic malware investigation. The use of static features makes it possible to extract important information about the file's compositional details. PE-section, import, header, byte, and Opcode histograms are commonly used for static malware analysis [2].

But given these traits, it's likely that important information on cutting-edge malware strategies—like obfuscation, metamorphism, mutation, and oligomorphic code—that are used to evade detection will be left out. Dynamic malware analysis may record the behaviours and activities of the executable file, which can then be utilised to identify and classify malware [3]. The feature set that is most commonly used in dynamic malware analysis is the API call sequencing feature. This is because it not only documents the binary's interactions with various system instances, but it also discloses the intent behind the virus's creation. Additionally, in order to more precisely identify weaknesses and boost efficiency, researchers used a technique called hybrid analysis, which leveraged an aggregation of static and dynamic features [4].

Deep learning has garnered a lot of attention and shown promise in the detection and classification of malware due to its ability to learn complex representations and patterns from complex data. Without the need for created features, deep learning models, such as CNNs and RNNs, may automatically learn hierarchical representations from raw data (such as opcode sequences, byte-level information, or binary code) [5]. This ability to automatically extract features from data has made it feasible to capture intricate viral patterns. Deep learning models may assess software behavioural patterns by examining series of system calls, API requests, or network data. Recurrent neural networks, in particular Long Short-Term Memory (LSTM) networks, have been used in malware analysis and classification utilising behavioural sequences. Deep learning techniques can combine static (file-based) and dynamic (behavior-based) analysis for comprehensive malware identification. Combining traits collected from static and dynamic analysis can lead to improved classification accuracy [6]. File headers and byte sequences are examples of static analysis features, whereas system actions and API requests are examples of dynamic analysis features. Researchers have looked into the use of deep learning algorithms to recognise malware and block its evasion strategies. Robustness techniques and adversarial training have been used to create models that are less likely to escape. Ensemble methods that combine domain adaptability, transfer learning techniques (pre-trained models), and deep learning architectures have been used in malware research. These techniques leverage information from large datasets or related fields to improve classification accuracy in scenarios where labelled data is scarce [7]. Integrating feature fusion approaches into machine learning frameworks is an intriguing and potentially profitable direction for malware classification [8]. This approach considers the complexity and diversity of malware's behaviours and characteristics. Feature fusion recognises that complete malware classification requires an integrated perspective integrating several properties from various sources, including behavioural traits, network traffic patterns, and file-based features. Combining these several traits provides a more comprehensive picture of malicious software, which improves classification models' resilience and accuracy [9]. Recent experiments with feature fusion and machine learning have shown promising improvements in the accuracy of malware categorization. These techniques raise detection rates while being impervious to malware evasive strategies. This work investigates the combination of cutting-edge machine learning models and feature fusion techniques, adding to the ongoing efforts to fortify cybersecurity and protect digital environments against new threats:

In this study, we combine feature fusion and machine learning to examine the latest advancements and trends in malware classification. Here's an explanation of feature fusion, machine learning, and malware classification. In this study, we explore how these approaches tackle contemporary cybersecurity concerns and offer experimental results. Researchers, cybersecurity professionals, and businesses searching for tactics and fixes to counter new malware threats may find great value in this study.

2. RELATED WORKS

Malware samples are analysed to find the characteristics that may be used to determine them. Machine learning approaches [10] state that as malware becomes more complex throughout its lifespan, understanding cryptic malware protection has become essential to malware detection. Furthermore, two forms of malware analysis are still often employed in the process of detecting potentially harmful apps [11]. ML-based malware detection algorithms analyse data through feature extraction. Machine learning techniques were employed using these characteristics (API calls, Assembly, and Binary) [12] to categorise malware.

In the topic of malware classification, several methods for recognising and categorising malware have been developed. Conventional approaches that rely on patterns, such signature-based detection, can fall short when faced with new and unexpected threats [13]. When employing heuristic-based algorithms to identify potential threats based on behavioural patterns, there is a potential for false alarms. The use of supervised, unsupervised, and deep learning techniques in machine learning to evaluate massive malware data sets and differentiate between safe and hazardous software has resulted in a significant shift. Moreover, behavioural analysis, which is done in controlled environments, can be used to identify dangerous behaviours [14]. Some advanced techniques use machine learning, behavioural analysis, heuristics, and signatures to create hybrid models that combine several detection techniques for increased accuracy [15]. Using dynamic analysis, which involves seeing malware in operation in a controlled environment in real-time, one may understand how malware impacts systems. Additionally, combining several data sets from various sources is a potent technique that offers a thorough picture of malware activities [16]. As the cybersecurity landscape shifts, ongoing research attempts to enhance and modify these strategies to address the always evolving cyber threats.

In contrast to conventional malware, which was only run once, current malware is more specialised, stealthy, and persistent [17]. Conventional malware was broad and easily accessible. Since zero-day infections make use of more recent vulnerabilities that have not yet been made public, they are challenging to identify [18]. Artificial Intelligence, machine learning, and deep learning techniques are being used in many computer science domains, from natural language processing to virus detection techniques. After researching Android malware, author [19] is currently developing a multi-feature consensus-based decision fusion adaptive identification component to use this malware (MCDF). Srndic et al. [20] used machine learning approaches in combination with static analysis to categorise malware samples. In this study, two distinct file formats were examined. Resource-draining executables are increasingly being included into PDF and SWF files by malware writers. In this study, 40,000 SWFs and 440,000 PDFs were examined. The design of this technology allowed for the detection of harmful code in Adobe PDF and Flash (SWF) files.

Signatures play a major role in an anti-virus or malware detection system's ability to recognise unusual activities. This method looks for certain viral patterns in a huge sample of signatures. The signature-based approach looks for disturbances by consulting a list of known assaults that has been previously provided. This configuration may detect malware in a wide range of applications, but in order to stay effective, the designated signature database has to be updated on a regular basis. Because adaptable malware is continually growing, it is therefore less successful in detecting hazardous exercises when utilising the signature-based technique [21]. The anti-virus provider uses heuristic methods that are able to recognise harmful software and handle their signatures [22].

Static feature extraction is done with feature extraction programmes like HashGenerator, PsStudio, PeView, and PeExplorer. Tools for disassembling code, such as Lida, Cpstone, and IDA Pro, are used to do static analysis at the code level. Static malware features such as Opcode [15], String [23] (‘APIcallname’, ‘mytime’, ‘kernal32’), and N-gram [23] (‘mail’, ‘ili,’ and ‘ftw’)('PUSH, ADD, SUB, MOV,'), hash values ('e5dadf6524624f79c3127e247f04b548'), PE For analysis, the header data [24] (field value, checksum, size, and symbol) is extracted. It may be possible to simplify the signature-based identification problem to a straightforward string matching problem. This basically means that it keeps searching over a large string collection for a pattern or substring. Approximately 45–75 percent of the computational time is devoted to this process alone [25]. In string matching, two of the most used algorithms are Aho-Corasick and Boyer-Moore. Even with the prevalence of unpacking techniques, de-obfuscating every malware component is a challenging task.

A malware detection approach for mobile phones based on an artificial immune system was proposed by WU Bin et al. (2015) [26]. To improve the accuracy of detection, a clone and mutation approach is used in addition to different detectors. It was also demonstrated that current features are particular instances of fuzzy token similarity. Token-based resemblance and character-based resemblance were joined to form a new similarity matrix. A signature-based method was created by Jiannan Wang et al. (2011) [27] to address the fuzzy-token similarity joins issue. It is discovered that the token-sensitive method performs better than other sig-nature strategies already in use. As an addition to the existing signature methods for edit distance, edit similarity was added. ScaleMalNet is a deep learning system for identifying zero-day malware that makes use of image processing, dynamic analysis, and static data, as recommended by the study in [8]. [28] proposed a strategy based on behavioural traits to define malware. They collect API call traces from malware samples in a controlled virtual environment and perform dynamic inspection on a dataset of typically early malware in order to have the suggested model eliminated. After the traces are analysed, more sophisticated features, or "actions," are produced. According to Arivudainambi, Varun, et al.'s methodologies [29], network traffic analysis may be used to identify malicious activity. Better anti-network traffic methodological strategies need the use of PCA. The suggested technique was evaluated by executing 1,000 malicious files in many sandboxes, including as Cuckoo, Limon, and Noriben. There was a 99 percent success rate in detecting malware using this strategy.

Anti-virus or malware detection systems rely heavily on signatures to identify anomalous activity. This approach looks for certain patterns in a vast collection of signatures in order to detect infections. The signature-based method looks for disturbances using a previously specified list of known attacks. The setup can identify malware in a variety of situations, but in order to keep functioning, it needs regular updates to the signature database that is given. The signature-based method's effectiveness in identifying harmful activities is restricted since adaptable malware is constantly evolving [21]. To properly identify malicious software and preserve signatures, the anti-virus provider use metaheuristic approaches [22].

programmes for extracting static features, such HashGenerator, PsStudio, PeExplorer, and PeView. Disassembler tools like Lida, Cpstone, and IDA Pro can be used for code-level static analysis. Some static elements, such N-grams, strings, opcodes, hash values, and PE header information, are retrieved in order to analyse the virus. If string matching turns out to be too challenging, signature-based identification can end up being quite simple. In real terms, this implies that it looks for patterns or substrings by sifting through a sizable string collection. Between 45 and 75 percent of the processing time is accounted for by this single procedure [25]. Well-known string-matching algorithms are Boyer-Moore and Aho-Corasick. Even with a number of unpacking techniques available, decrypting every piece of malware is still a difficult task. A malware classifier that could handle polymorphic patterns was developed by Narayanan et al. (2016) [30] using supervised machine learning techniques.

There are two methods in which anomalies are incorporated into behavior-based techniques. An anomaly is a malfunction brought on by malicious files. Files that exhibit unusual behaviour that deviates from the behaviour of regular files stored are considered malicious.

This section goes over behavioral-based malware detection techniques in depth. The use of sophisticated techniques to detect malware is mentioned. Bailey et al. (2007) [31] proposed one such technique that captured the API calls made by malware. In order to maintain appropriate precision while accounting for both dynamic and static inquiry points of interest, Eskandari et al. (2013) [32] introduced a novel hybrid approach called HDM-Analyzer. As a result, HDM-Analyzer is able to predict that there is minimal performance deterioration because the majority of the core leadership is based on actual data. Sheen and colleagues (2015) [19] created MCDF. By integrating the classifiers' choices using the collective technique based on the probability hypothesis, which is utilised to form a group of classifiers, malicious record characteristics such as the consent-based features and API call-based features are examined to deliver a better finding.

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)

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]. The algorithmic conversions for machine learning models entail initializing and training each classifier and making predictions accordingly. For Random Forest (RF), the scikit-learn library is employed to instantiate a Random Forest Classifier, which is then trained on the training data. Predictions are generated using the trained model. XGBoost implementation involves converting the data into DMatrix format, setting appropriate parameters for multi-class classification, and training the model using the xgboost.train function. K-Nearest Neighbors (KNN) classifier is instantiated using KNeighborsClassifier, trained on the training data, and used to predict labels for the test data. For Decision Tree (DT), a DecisionTreeClassifier is initialized, trained on the training data, and utilized for generating predictions. These implementations enable the application of Random Forest, XGBoost, KNN, and Decision Tree algorithms for classification tasks, providing versatility in modeling approaches for different datasets and problem domains. After a thorough study, the machine learning 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 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 ML models, optimizes hyperparameters, evaluates performance across different feature sets, and finally uses a selected model for both the fused feature set and individual feature sets.

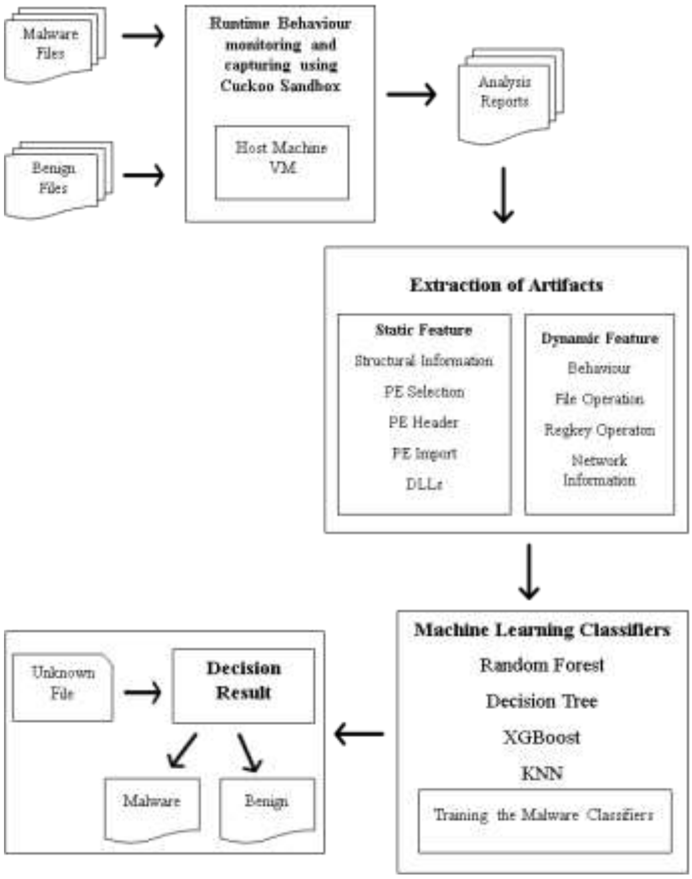

Fig.1. Proposed classification scheme

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

Input:

PE_section: List of PE section features (ps1, ps2, ..., psm)

PE_import: List of PE import features (pi1, pi2, ..., pin)

PE_API: List of PE API features (pa1, pa2, ..., pap)

PE_image: List of PE im features (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) to create a fusion feature = {F1, F2, ..., Ft}, where m + n + p + q = t.

Preprocessing:

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

end for

4. EXPERIMENTAL RESULTS

Different algorithms, including Random Forest, XGBoost, Decision Tree, and KNN, were utilised in order to evaluate a multi-view feature fusion strategy for successful malware detection. In particular, the use of the top 40 characteristics in conjunction with Random Forest demonstrated the best accuracy, which was 97.31%.

The Dynamic Analysis, on the other hand, comprised evaluating several feature subsets (top 10, 20, 30, 40, 50, 60, and 70) with the same algorithms. The results showed that using Random Forest with the top 40 features resulted in the greatest accuracy of 99.11%. The Hybrid Analysis, which combines forty static characteristics with sixty dynamic features, displayed remarkable 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	96.5433	95.7507	95.5085	93.5270
20	96.8516	96.12505	95.9709	93.8132
30	97.2258	96.67547	96.0149	93.7912
40	97.3139	97.09379	96.2351	93.8353
50	97.1818	97.00572	96.367	93.8132

The effectiveness of the feature fusion methodology was demonstrated by the fact that the multi-feature approach was found to be superior than the use of a single feature through comparison. With regard to the specifics, it attained an accuracy of 97.31% for static analysis, 99.11% for dynamic analysis, and an astounding 99.64% for hybrid analysis. Despite the fact that these accomplishments are being highlighted, it is essential to realise both the advantages and the limits of these techniques in the categorization of various types of malware. The multi-feature method displays significant advances in accuracy; yet, the field may still encounter obstacles in specific cases, such as by employing evasion strategies that are used by malevolent actors.

Table.3. Dynamic Analysis with feature numbers and algorithms

Feature_no.	Random _forest	XGBoost	DecisionTree	KNN
10	96.52135	95.9709	96.2791	94.2536
20	98.5909	98.2826	97.8643	95.8388
30	98.7010	98.9431	98.2386	90.2245
40	99.0973	99.0752	98.4588	90.8190
50	99.0312	99.0312	98.4368	90.4007
60	98.8331	99.1193	98.5689	90.2906
70	99.0092	99.0752	98.3927	90.2906

Nevertheless, this exhaustive study demonstrates the possibility and usefulness of employing a wide variety of characteristics and models for the purpose of improving malware categorization.

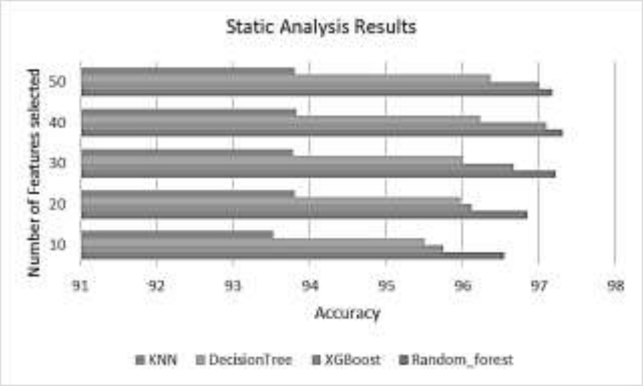

Fig.2. Accuracy in static analysis with the number of features set selected

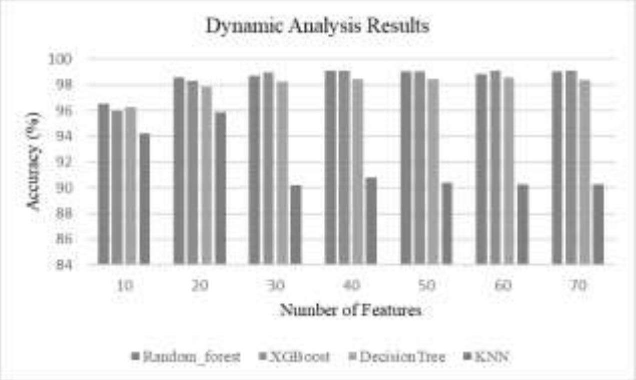

Fig.3. Accuracy in dynamic analysis with the number of features set selected

Figure 4 displays the findings of Random Forest, which used the top 40 features and reached an accuracy of 97.31%. Figure 5 illustrates that XGBoost, which used the top 60 features, achieved the best accuracy of 99.11% for the dynamic analysis. For hybrid analysis, the top 40 static and 60 dynamic features were chosen. The Random_forest, XGBoost, DecisionTree, and KNN algorithms were found to have an accuracy of

99.64773227653016, 99.88991633641568, 99.09731395860855, and 93.83531483927786 respectively. When it comes to the categorization of malware, the multi-feature method that has been offered yields superior results when compared to the use of a single feature. As a result, the approach of feature fusion was able to reach an accuracy of 97.31% for static analysis and 99.11% for dynamic analysis. A hybrid analysis also obtains a 99.64% success rate. While taking into consideration the sequential_1 model, the number of epochs that were used was 100, and the training outcomes revealed a loss of 0.0398 and an accuracy of 0.9868 percent. An accuracy of 99.18% is supported by the precision recall f1-score. The results of the testing with epoch 100 using the sequential_4 model with a total of 711,342 reveal that the accuracy is 75.17%..

5. CONCLUSION

In conclusion, the multi-view feature fusion technique has a significant potential to enhance the accuracy of malware classification systems. Methods for classifying malware that rely solely on static, dynamic, or behavior-based features have, traditionally speaking, been unable to capture the entire complexity of malware, which has resulted in findings that are less than optimal. Experimental findings that utilised the Random Forest, XGBoost, Decision Tree, and KNN algorithms demonstrated notable successes. These algorithms were utilised in the experiments. Utilisation of the top forty characteristics resulted in an accuracy rate of 99.11% for dynamic analysis and a rate of 97.31% for static analysis. A hybrid analysis that merged forty static features with sixty dynamic features resulted in an accuracy rate of 99.64%. The fact that these results exhibit considerable increases in classification accuracy across a variety of trials demonstrates that the strategy of feature fusion is more successful than applying individual characteristics separately. It is essential to make use of Deep Learning or Machine Learning models that are capable of overcoming evasion strategies in order to make malware detection systems more resilient. This is because malicious actors may change binary files in order to protect themselves from detection. It is necessary to do further research and make significant advancements in the field of model construction in order to combat emerging threats and improve the overall efficiency of classified malware techniques.

In the future, the classification of malware will undergo substantial improvements in a number of significant elements. We could discover ways to increase the interpretability of models, examine more complicated deep learning architectures, fortify machine learning models against adversarial assaults, and boost feature engineering in order to give more informative and robust feature sets. These are just some of the potential outcomes. For the efficient processing of large datasets and the detection of threats in real time, scalability and breakthroughs in unsupervised learning methods are absolutely necessary. The protection of user privacy, the improvement of behavioural analysis, and the promotion of collaborative defensive systems are all essential components for the comprehensive identification of malware. It is necessary to develop and maintain datasets on a consistent basis in order to keep things moving ahead and to ensure that future malware classification algorithms are robust. In the event that these concern areas are addressed, the capability of the field to combat evolving cyber threats will be significantly improved.

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] W. Han, J. Xue, Y. Wang, L. Huang, Z. Kong, and L. Mao, “MalDAE: Detecting and explaining malware based on correlation and fusion of static and dynamic characteristics,” Computers and Security, vol. 83, pp. 208–233, 2019, doi: 10.1016/j.cose.2019.02.007.

[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] 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.

[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] M. Ashik et al., "Detection of malicious software by analyzing distinct artifacts using machine learning and deep learning algorithms," Electronics (Switzerland), vol. 10, no. 14, pp. 1–28, 2021, doi: 10.3390/electronics10141694.

[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.