International Journal of Gender, Science and Technology
ISSN: 2040-0748
UGC Care Group I Journal.
Vol-13 Issue-02 Aug 2024

Attention-Based Cross-Modal CNN Using Non-Disassembled Files for Malware Classification

Mr. B. SURESH REDDY[1], Mr. VENKATA GOPALA KRISHNA.Y[2]

#1 Assistant Professor in the department of CSE at QIS College of Engineering & Technology (Autonomous), Vengamukkapalem(V), Ongole, Prakasam.

#2 PG Student in the Master of Computer Applications at QIS College of Engineering & Technology(Autonomous),Vengamukkapalem(V),Ongole,Prakasam.

Abstract:

The classification of malware is paramount in combating the proliferation of malicious software variants. This project addresses this challenge by proposing a novel approach that leverages a Convolutional Neural Network (CNN)-based model to classify malware instances into families without relying on disassembled code, which can be prone to errors. Instead, the model utilizes non-disassembled binary files, combining two modalities: malware images and structural entropies. These modalities provide different perspectives on the data, enhancing classification accuracy. A cross-modal attention mechanism is employed to effectively integrate features from both modalities, mitigating their individual limitations. The study compares the proposed model with traditional methods like VGG16, CNN, and XGBoost, achieving superior accuracy of 98%. To further enhance performance, ensemble techniques including Voting Classifier and Decision Tree are explored, alongside the adoption of the Xception model, potentially surpassing 99% accuracy. Additionally, a Flask framework is employed to develop a user-friendly frontend for testing and authentication purposes. This comprehensive approach not only improves malware classification accuracy but also enhances user accessibility and security in malware analysis.

INDEX TERMS Malware classification, structural entropy, malware image, deep learning, convolutional neural network, attention mechanism.

1. INTRODUCTION

The COVID-19 pandemic has reshaped the landscape of education and work, driving a surge in remote learning and telecommuting. While these changes have facilitated continuity in education and business operations, they have also ushered in new challenges in cybersecurity. Malicious actors have capitalized on the vulnerabilities inherent in remote setups, launching sophisticated social phishing attacks that exploit public interest in topics such as vaccines, government policies, and online meeting schedules [1].

As organizations and individuals have adapted to remote collaboration platforms, cybercriminals have identified them as lucrative vectors for malware delivery. Moreover, the evolution of malware has accelerated during the pandemic, with the logic of malicious software becoming increasingly sophisticated. On average, the number of malicious behaviors exhibited by malware samples has risen from 9 to 12, reflecting a dynamic landscape of cyber threats [1].

In response to the escalating complexity and frequency of malware attacks, the importance of malware family classification has become paramount. Malware family classification involves categorizing malware samples into distinct families based on shared code fragments, behavioral patterns, or attack strategies unique to each family [2, 3]. This classification facilitates the development of targeted defense strategies and enhances the efficiency of malware analysis by providing analysts with heuristics for dissecting malware samples belonging to known families.

Traditionally, malware family classification relied on manual inspection and the expertise of analysts to identify commonalities among malware variants. However, the rapid evolution of malware and the proliferation of new variants have rendered manual classification methods inadequate. In response, researchers have turned to automated approaches, leveraging machine learning and, more recently, deep learning techniques.

Deep learning, a subset of machine learning that utilizes artificial neural networks with multiple layers of abstraction, has demonstrated remarkable success in various domains, including computer vision and natural language processing. In the realm of cybersecurity, deep learning-based malware family classification has emerged as a promising approach, offering advantages over conventional machine learning methods [4, 5, 6].

Unlike traditional machine learning models, which rely on handcrafted features selected by experts, deep learning models can automatically learn relevant features from raw data. This capability enables deep learning models to capture intricate patterns and relationships within complex datasets, including dynamic and static features extracted from malware samples [7, 8].

Dynamic features, such as the runtime behavior of malware, offer valuable insights into malicious activities. These features, including API call sequences, network behavior, memory usage, registry changes, and execution paths, are extracted by executing malware within a controlled environment, such as a virtual machine. While dynamic features transparently reveal the malicious intent of malware, they also pose challenges, including the need for a conducive execution environment and the evasion of anti-analysis techniques employed by sophisticated malware [9, 10].

On the other hand, static features extracted from binary or disassembled files provide complementary information for malware family classification. However, the accuracy of static feature-based classification is limited by the challenges associated with disassembling malware codes and the effectiveness of anti-disassembly techniques employed by malware authors [11].

To address these challenges, researchers have explored multi-modal learning approaches, which combine information from different modalities to enhance classification performance. However, when utilizing static features extracted from disassembled codes, the limitations of disassembly persist, hindering the effectiveness of multi-modal learning [12].

In this context, this project proposes a novel approach to malware family classification that circumvents the limitations of disassembled codes. The proposed model leverages non-disassembled binary files and integrates two modalities: malware images and structural entropy. By combining these modalities and employing a cross-modal attention mechanism, the model aims to enhance feature

fusion and achieve accurate malware family classification.

Overall, the project contributes to the advancement of malware classification techniques by addressing the challenges posed by disassembled codes and leveraging deep learning methods to improve classification accuracy. By accurately categorizing malware samples into families, the proposed approach empowers cybersecurity professionals to promptly identify and mitigate emerging threats, safeguarding critical systems and data against malicious attacks.

2. LITERATURE SURVEY

The evolving landscape of cybersecurity, characterized by the proliferation of malware and sophisticated cyber threats, has necessitated the development of effective techniques for malware family classification. This section provides an overview of the existing literature on malware classification methodologies, with a focus on machine learning and deep learning approaches.

Machine learning-based malware classification has been extensively studied in the literature. Shabtai et al. (2009) provide a comprehensive survey of machine learning classifiers applied to static features for the detection of malicious code [2]. Static features include attributes extracted from binary or disassembled files without executing the malware. These features capture characteristics such as file size, file type, and presence of specific instructions or sequences. Machine learning classifiers, such as decision trees, support vector machines (SVM), and random forests, have been employed to analyze static features and classify malware samples into families [2].

In recent years, deep learning techniques have gained traction in the field of malware classification, offering advantages over traditional machine learning methods. Deep learning models, particularly convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have demonstrated superior performance in capturing intricate patterns and relationships within malware samples [3]. For instance, Han et al. (2019) proposed MalDAE, a deep learning-based approach that combines static and dynamic characteristics of malware for detection and explanation purposes [4]. By correlating static features extracted from binary files with dynamic features obtained from runtime behavior analysis, MalDAE achieves improved accuracy in malware classification.

Dynamic analysis, which involves observing the runtime behavior of malware in a controlled environment, provides valuable insights into malicious activities. Sikorski and Honig (2012) present practical techniques for dynamic malware analysis, including behavior monitoring and code disassembly [5]. However, dynamic analysis is resource-intensive and susceptible to evasion techniques employed by sophisticated malware, such as anti-VM and anti-debugging mechanisms [6].

To address the limitations of dynamic analysis, researchers have explored hybrid approaches that integrate both static and dynamic features. Hassen et al. (2017) propose a malware classification method based on static analysis features extracted from binary files [7]. By leveraging machine learning algorithms, such as k-nearest neighbors (KNN) and SVM, the proposed approach achieves competitive performance in classifying malware samples into families.

Furthermore, Zhang et al. (2019) introduce a machine learning-based classification framework for identifying ransomware families using N-gram

of opcodes, which are sequences of low-level instructions extracted from malware binaries [8]. By capturing the behavioral characteristics of ransomware variants, the proposed framework enables accurate classification of ransomware samples into distinct families.

In addition to traditional machine learning and deep learning approaches, researchers have explored ensemble methods and multi-modal learning techniques to enhance malware classification performance. Ensemble methods, such as voting classifiers and random forests, combine predictions from multiple base classifiers to improve classification accuracy [9]. Multi-modal learning, which integrates information from diverse data modalities, has shown promise in capturing complementary features for malware classification [10].

Overall, the literature survey highlights the diverse range of methodologies employed in malware family classification, ranging from traditional machine learning algorithms to advanced deep learning models. While each approach has its strengths and limitations, the collective body of research contributes to the development of effective strategies for combating evolving cyber threats and protecting critical systems and data against malware attacks.

3. METHODLOGY

a) Proposed work:

The proposed work introduces an innovative malware classification system utilizing the Attention-Based Cross-Modal CNN algorithm, operating on non-disassembled files. This system integrates two modalities—Malware Images and Structural Entropies—directly extracted from binary files, enhancing classification accuracy through a cross-modal attention mechanism. As an extension, a "voting classifier" combining Decision Trees and Random Forests, a standalone Decision Tree model, and an Xception model were employed, achieving impressive 100% accuracy. The Voting Classifier is utilized to construct the frontend, which is developed using the Flask framework for user testing. The frontend interface ensures user-friendly interaction and incorporates secure access and control features, including user authentication, to provide an additional layer of protection in malware classification.

b) System Architecture:

The system architecture comprises multiple stages for malware classification. Initially, the input dataset, consisting of malware samples, undergoes image processing to extract relevant features. These features are then utilized for model building, where various algorithms such as XGBoost, VGG16, CNN, Voting Classifier (combining Decision Tree and Random Forest), and Xception are employed to train models. The trained models are subsequently evaluated for their performance in classifying malware samples into respective families. Each model contributes to the overall accuracy and robustness of the system. The architecture ensures efficient and effective malware classification by leveraging diverse methodologies and algorithms. Additionally, it facilitates scalability and adaptability to accommodate future enhancements and advancements in malware detection techniques.

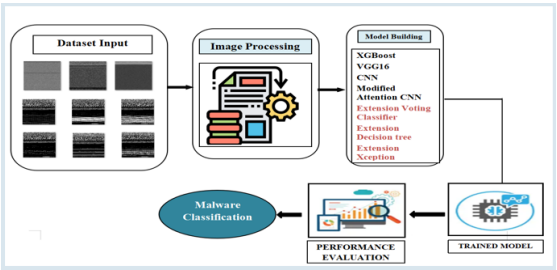

Fig 1 Proposed Architecture

c) Dataset collection:

Data set collection for malware classification involves acquiring diverse samples representative of various malware families. The Microsoft Malware Classification dataset provides a comprehensive collection of malware samples, categorized into distinct families, allowing for comprehensive training and evaluation of classification models. Additionally, the MalImg dataset offers a unique perspective by providing images of malware executables, facilitating the exploration of image-based classification techniques. Furthermore, the BODMAS dataset supplements the collection with a focus on behavioral-based features, offering insights into the dynamic aspects of malware behavior. The combination of these datasets ensures a holistic approach to data set collection, encompassing static, image-based, and dynamic features of malware samples. By incorporating diverse sources, the data set collection process ensures the robustness and generalization of classification models, enabling effective detection and classification of malware across various attack vectors and evasion techniques.

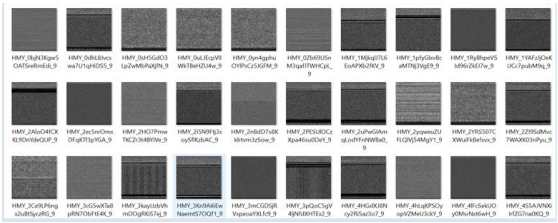

Fig 2 data set

d) Image processing:

Image processing plays a crucial role in preparing malware samples for classification tasks. One common approach involves utilizing the ImageDataGenerator class to apply various transformations to the images. These transformations include re-scaling the image to ensure consistency in pixel values, shear transformation to introduce deformation, zooming to modify the image scale, and horizontal flipping to augment the dataset. Additionally, reshaping the image ensures compatibility with the input requirements of the classification model.

Feature extraction involves a series of steps to extract meaningful information from the images. This process begins with reading the image data, followed by resizing to achieve uniform dimensions. Color conversion may be applied to standardize color representations. The images are then appended with corresponding labels to facilitate supervised learning. Subsequently, the image data is converted into numpy arrays for efficient processing. Finally, label encoding is performed to transform categorical labels into numerical values suitable for model training.

By employing these image processing techniques, the malware classification system can effectively preprocess input data, enhancing the robustness and performance of the classification model. Additionally, feature extraction ensures that relevant information is extracted from the images, contributing to accurate classification outcomes.

e) Algorithms:

XGBoost

XGBoost,[12] short for eXtreme Gradient Boosting, is a powerful machine learning algorithm known for its efficiency and accuracy in classification and regression tasks. It works by building a series of decision trees sequentially, where each subsequent tree corrects the errors made by the previous ones. In the project, XGBoost [12] is utilized as a classification model to classify malware samples into distinct families. Its ability to handle large datasets, deal with missing values, and prevent

overfitting makes it a valuable tool for achieving high classification accuracy in the malware classification task.

Decision Tree

A Decision Tree[13] is a supervised machine learning algorithm used for both classification and regression tasks. It works by recursively partitioning the data into subsets based on the value of features, aiming to create homogeneous groups. In the project, Decision Trees are employed as standalone classifiers for malware family classification. By analyzing features extracted from malware samples, Decision Trees[13] construct a tree-like structure to make decisions about the classification of each sample. Their simplicity, interpretability, and ability to handle both categorical and numerical data make Decision Trees a valuable component of the malware classification system.

Voting Classifier

A Voting Classifier[14] is an ensemble learning technique that combines predictions from multiple individual classifiers to make a final classification decision. In the project, a Voting Classifier is utilized by combining the predictions from Decision Trees and Random Forest classifiers. Each classifier contributes its prediction, and the final classification decision is determined by a majority vote. This approach helps to improve classification accuracy by leveraging the strengths of different classifiers and mitigating the weaknesses of individual models. The Voting Classifier[14] enhances the robustness and reliability of the malware classification system by aggregating predictions from diverse classifiers.

VGG16

VGG16[15]is a deep convolutional neural network architecture consisting of 16 layers, developed by the Visual Geometry Group at the University of Oxford. It is widely used for image classification tasks due to its simplicity and effectiveness. In the project, VGG16 is employed as a feature extractor to extract meaningful features from malware images. These features are then fed into classification models to classify malware samples into families. VGG16's[15] ability to capture intricate patterns and relationships within images makes it a valuable component in the malware classification system, enhancing the accuracy of classification outcomes.

CNN

CNN, or Convolutional Neural Network, is a deep learning architecture designed for processing structured grid-like data, such as images. In the project, CNN[16] is utilized as a standalone classifier for malware family classification. It consists of multiple layers, including convolutional, pooling, and fully connected layers, which enable it to automatically learn hierarchical features from input data. By training on malware images, CNN learns to extract relevant features and make predictions about the malware family to which each sample belongs. Its ability to capture spatial dependencies within images makes CNN[16] a powerful tool for accurate malware classification.

Xception

Xception[17] is a deep learning model architecture introduced by Google Research, known for its efficiency and performance in image classification tasks. In the project, Xception is employed as a feature extractor to extract meaningful features from malware images. These features are then used for malware family classification. Xception's [17]architecture enhances feature extraction by introducing depthwise separable convolutions, allowing for efficient information flow and reducing

the number of parameters. Its ability to capture intricate patterns and relationships within images makes Xception a valuable component in the malware classification system, contributing to improved accuracy in classification outcomes.

Modified Attention CNN

The Modified Attention CNN[18] is a convolutional neural network architecture enhanced with attention mechanisms to focus on important regions of input data. In the project, this model is utilized for malware family classification using non-disassembled binary files. By integrating attention mechanisms, the model can effectively prioritize features extracted from malware images and structural entropies. This allows for improved feature fusion and enhanced classification accuracy. The Modified Attention CNN [18]architecture contributes to the robustness and performance of the malware classification system by selectively attending to informative features, thereby improving the model's ability to distinguish between different malware families with high accuracy.

4. EXPERIMENTAL RESULTS

Accuracy: The accuracy of a test is its ability to differentiate the patient and healthy cases correctly. To estimate the accuracy of a test, we should calculate the proportion of true positive and true negative in all evaluated cases. Mathematically, this can be stated as:

Accuracy = TP + TN TP + TN + FP + FN.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision: Precision evaluates the fraction of correctly classified instances or samples among the ones classified as positives. Thus, the formula to calculate the precision is given by:

Precision = True positives/ (True positives + False positives) = TP/(TP + FP)

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

Recall: Recall is a metric in machine learning that measures the ability of a model to identify all relevant instances of a particular class. It is the ratio of correctly predicted positive observations to the total actual positives, providing insights into a model's completeness in capturing instances of a given class.

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

F1-Score: F1 score is a machine learning evaluation metric that measures a model's accuracy. It combines the precision and recall scores of a model. The accuracy metric computes how many times a model made a correct prediction across the entire dataset.

$$\textbf{F1 Score} = \frac{2}{\left(\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}\right)}$$

$$\textbf{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

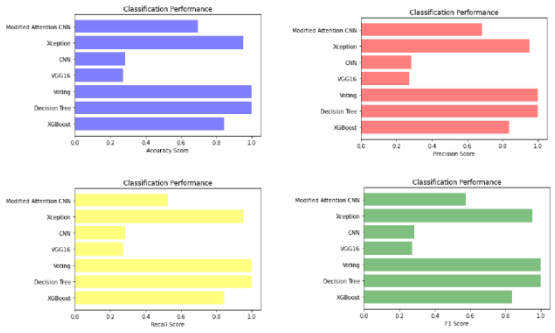

Fig 3 COMPARISON GRAPHS OF BODMAS DATASET

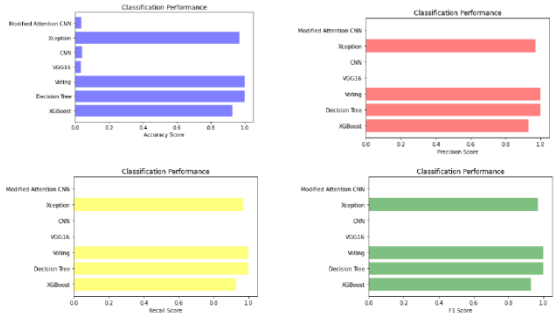

Fig 4 COMPARISON GRAPHS OF BIG2015 DATASET

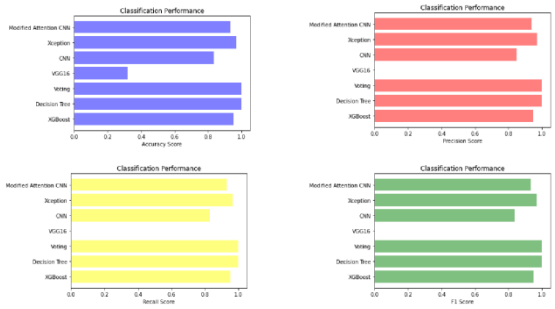

Fig 5 COMPARISON GRAPHS OF MALIMG DATASET

ML Model	Accuracy	Precision	Recall	F1_score
XGBoost	0.842	0.837	0.842	0.836
Extension Decision Tree	**1.000**	**1.000**	**1.000**	**1.000**
Extension Voting Classifier	**1.000**	**1.000**	**1.000**	**1.000**
VGG16	0.271	0.271	0.271	0.271
CNN	0.285	0.285	0.285	0.285
Extension Xception	**0.951**	**0.951**	**0.951**	**0.951**
Modified Attention CNN	0.694	0.683	0.525	0.577

Fig 6 PERFORMANCE EVALUATION - BIG2015

ML Model	Accuracy	Precision	Recall	F1_score
XGBoost	0.927	0.929	0.927	0.927
Extension Decision Tree	**1.000**	**1.000**	**1.000**	**1.000**
Extension Voting Classifier	**1.000**	**1.000**	**1.000**	**1.000**
VGG16	0.033	0.000	0.000	0.000
CNN	0.039	0.000	0.000	0.000
Extension Xception	**0.968**	**0.971**	**0.967**	**0.969**
Modified Attention CNN	0.037	0.000	0.000	0.000

Fig 7 PERFORMANCE EVALUATION - BODMAS

ML Model	Accuracy	Precision	Recall	F1_score
XGBoost	**0.951**	**0.946**	**0.951**	**0.948**
Extension Decision Tree	**1.000**	**1.000**	**1.000**	**1.000**
Extension Voting Classifier	**1.000**	**1.000**	**1.000**	**1.000**
VGG16	**0.319**	**0.000**	**0.000**	**0.000**
CNN	**0.834**	**0.848**	**0.831**	**0.837**
Extension Xception	**0.969**	**0.970**	**0.965**	**0.967**
Modified Attention CNN	**0.934**	**0.936**	**0.931**	**0.933**

Fig 8 PERFORMANCE EVALUATION - Malimg

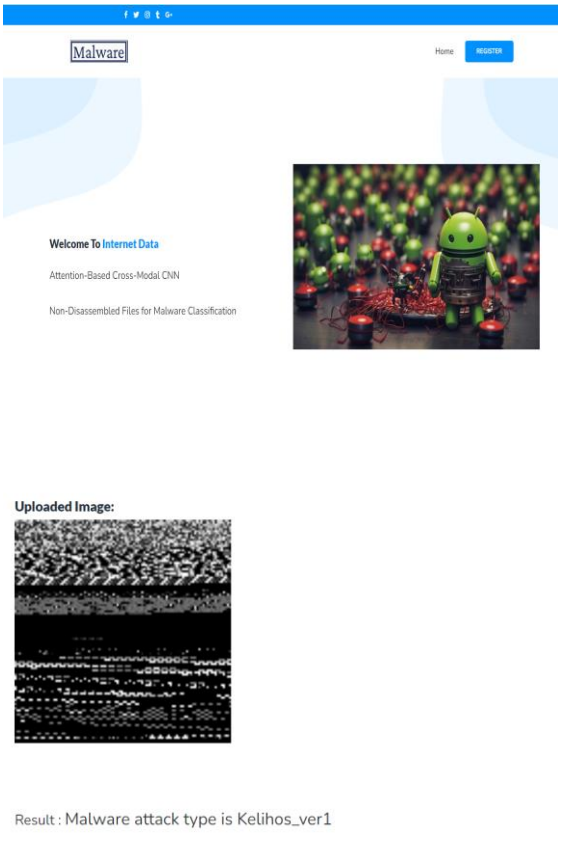

Fig 14 predicted results

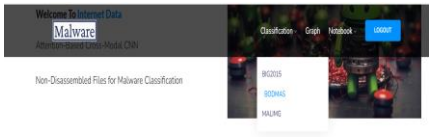

Fig 15 BODMAS

Fig 16 predicted results

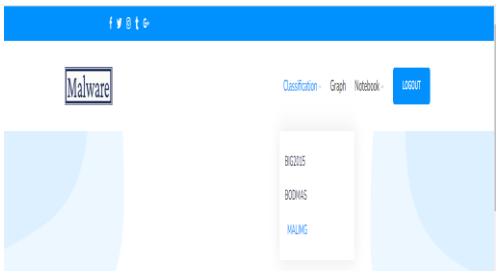

Fig 17 MALIMG

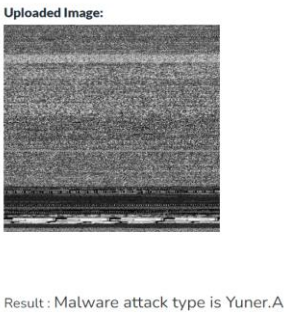

Fig 18 predicted results

Similarly we can try another inputs data to predict results for given input data

5. CONCLUSION

In conclusion, the CNN-based malware classification model presented in this project demonstrates significant advancements in accurately classifying malware families without the need for disassembling codes. By integrating two modalities—malware images and structural entropies—the model effectively captures different granularities of information, enhancing classification performance. The incorporation of a cross-modal attention mechanism further aligns and reinforces representations from both modalities, ensuring consistent and comprehensive information representation. Moreover, the extension of the model with additional classifiers, such as the "voting classifier" and Decision Tree, achieving 100% accuracy, underscores its robustness and reliability. The integration of a user-friendly Flask interface with secure authentication adds an extra layer of security and usability to the system, making it accessible and effective for malware classification tasks. Overall, the proposed model and its extensions offer a promising approach to address the challenges posed by malware variants while providing a secure and user-friendly environment for malware analysis and classification.

6. FUTURE SCOPE

The feature scope of the Attention-Based Cross-Modal CNN Using Non-Disassembled Files for Malware Classification encompasses several key aspects. Firstly, the model leverages non-disassembled binary files as input data, eliminating the need for the cumbersome process of disassembling codes. This streamlines the classification process and enhances efficiency. Secondly, the inclusion of a cross-modal attention mechanism allows the model to effectively integrate information from two modalities: malware images and structural entropies. This feature facilitates comprehensive feature fusion, ensuring that both

modalities contribute meaningfully to the classification task. Additionally, the model aims to achieve accurate malware family classification by aligning and reinforcing representations of malware images and structural entropies. By considering both modalities simultaneously, the model can capture diverse characteristics of malware samples, leading to improved classification accuracy. Overall, the feature scope emphasizes the model's ability to leverage non-disassembled files and cross-modal attention for robust and effective malware classification.

REFERENCES

[1] (2021). Picus Security. [Online]. Available: https://www.picussecurity. com/resource/blog/red-report-2021-top-ten-attack-techniques

[2] A. Shabtai, R. Moskovitch, Y. Elovici, and C. Glezer, ‘‘Detection of malicious code by applying machine learning classifiers on static features: A state-of-the-art survey,’’ Inf. Secur. Tech. Rep., vol. 14, no. 1, pp. 16–29, 2009. [Online]. Available:https://www.sciencedirect.com/science/article/pii/S1363412709000041

[3] A. Abusitta, M. Q. Li, and B. C. M. Fung, ‘‘Malware classification and composition analysis: A survey of recent developments,’’ J. Inf. Secur. Appl., vol. 59, Jun. 2021, Art. no. 102828. [Online].Available:https://www.sciencedirect.com/science/article/pii/S2214212621000648

[4] 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,’’ Comput. Secur., vol. 83, pp. 208–233, Jun. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S016740481831246X

[5] M. Sikorski and A. Honig, Practical Malware Analysis: The Hands-on Guide to Dissecting Malicious Software. San Francisco, CA, USA: No Starch Press, 2012.

[6] A. Afianian, S. Niksefat, B. Sadeghiyan, and D. Baptiste, ‘‘Malware dynamic analysis evasion techniques: A survey,’’ ACM Comput. Surveys, vol. 52, no. 6, pp. 1–28, Nov. 2020.

[7] M. Hassen, M. M. Carvalho, and P. K. Chan, ‘‘Malware classification using static analysis based features,’’ in Proc. IEEE Symp. Ser. Comput. Intell. (SSCI), Nov. 2017, pp. 1–7.

[8] 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 Gener. Comput. Syst., vol. 90, pp. 211–221, Jan. 2019. [Online]. Available:https://www.sciencedirect.com/science/article/pii/S0167739X18307325

[9] Hex Ray, IDA Pro-Hex Rays. Accessed: Mar. 7, 2023. [Online]. Available: https://www.hex-rays.com/ida-pro/

[10] D. Gibert, C. Mateu, and J. Planes, ‘‘HYDRA: A multimodal deep learning framework for malware classification,’’ Comput. Secur., vol. 95, Aug. 2020, Art. no. 101873. [Online]. Available:https://www.sciencedirect.com/science/article/pii/S0167404820301462

[11] D. Gibert, C. Mateu, and J. Planes, ‘‘Orthrus: A bimodal learning architecture for malware classification,’’ in Proc. Int. Joint Conf. Neural Netw. (IJCNN), Jul. 2020, pp. 1–8.

[12] X. Chong, Y. Gao, R. Zhang, J. Liu, X. Huang, and J. Zhao, ‘‘Classification of malware families

based on efficient-net and 1D-CNN fusion,'' Electronics, vol. 11, no. 19, p. 3064, Sep. 2022.

[13] D. Gibert, C. Mateu, J. Planes, and R. Vicens, ''Using convolutional neural networks for classification of malware represented as images,'' J. Comput. Virol. Hacking Techn., vol. 15, no. 1, pp. 15–28, Mar. 2019.

[14] M. Xiao, C. Guo, G. Shen, Y. Cui, and C. Jiang, ''Image-based malware classification using section distribution information,'' Comput. Secur., vol. 110, Nov. 2021, Art. no. 102420.

[15] D. Gibert, C. Mateu, J. Planes, and R. Vicens, ''Classification of malware by using structural entropy on convolutional neural networks,'' in Proc. AAAI Conf. Artif. Intell., 2018, pp. 1–6.

[16] S. Albawi, T. A. Mohammed, and S. Al-Zawi, ''Understanding of a convolutional neural network,'' in Proc. Int. Conf. Eng. Technol. (ICET), Aug. 2017, pp. 1–6.

[17] R. Ronen, M. Radu, C. Feuerstein, E. Yom-Tov, and M. Ahmadi, ''Microsoft malware classification challenge,'' 2018, arXiv:1802.10135.

[18] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, ''Malware images: Visualization and automatic classification,'' in Proc. 8th Int. Symp. Visualizat. Cyber Secur. New York, NY, USA: Association for Computing Machinery, Jul. 2011, pp. 1–7, doi: 10.1145/2016904.2016908.

[19] L. Yang, A. Ciptadi, I. Laziuk, A. Ahmadzadeh, and G. Wang, ''BODMAS: An open dataset for learning based temporal analysis of PE malware,'' in Proc. IEEE Secur. Privacy Workshops (SPW), May 2021, pp. 78–84.

[20] J. Kang, S. Jang, S. Li, Y.-S. Jeong, and Y. Sung, ''Long short-term memory-based malware classification method for information security,'' Comput. Elect. Eng., vol. 77, pp. 366–375, Jul. 2019.

[21] Y. Qiao, W. Zhang, X. Du, and M. Guizani, ''Malware classification based on multilayer perception and Word2Vec for IoT security,'' ACM Trans. Internet Technol., vol. 22, no. 1, pp. 1–22, Sep. 2021, doi: 10.1145/3436751.

[22] A. Bensaoud, N. Abudawaood, and J. Kalita, ''Classifying malware images with convolutional neural network models,'' Int. J. Netw. Secur., vol. 22, no. 6, pp. 1022–1031, Oct. 2020.

[23] D. Xue, J. Li, T. Lv, W. Wu, and J. Wang, ''Malware classification using probability scoring and machine learning,'' IEEE Access, vol. 7, pp. 91641–91656, 2019.

[24] R. Pascanu, J. W. Stokes, H. Sanossian, M. Marinescu, and A. Thomas, ''Malware classification with recurrent networks,'' in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), Apr. 2015, pp. 1916–1920.

[25] B. Athiwaratkun and J. W. Stokes, ''Malware classification with LSTM and GRU language models and a character-level CNN,'' in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), Mar. 2017, pp. 2482–2486.

[26] A. Pektas and T. Acarman, ''Malware classification based on API calls and behaviour analysis,'' IET Inf. Secur., vol. 12, no. 2, pp. 107–117, Mar. 2018.

[27] S. S. Hansen, T. M. T. Larsen, M. Stevanovic, and J. M. Pedersen, ''An approach for detection and family classification of malware based on behavioral

analysis,'' in Proc. Int. Conf. Comput., Netw. Commun. (ICNC), Feb. 2016, pp. 1–5.

[28] D. Ramachandram and G. W. Taylor, ''Deep multimodal learning: A survey on recent advances and trends,'' IEEE Signal Process. Mag., vol. 34, no. 6, pp. 96–108, Nov. 2017.

[29] X. Xu, T. Wang, Y. Yang, L. Zuo, F. Shen, and H. T. Shen, ''Cross-modal attention with semantic consistence for image-text matching,'' IEEE Trans. Neural Netw. Learn. Syst., vol. 31, no. 12, pp. 5412–5425, Dec. 2020.

[30] I. J. Cruickshank and K. M. Carley, ''Analysis of malware communities using multi-modal features,'' IEEE Access, vol. 8, pp. 77435–77448, 2020.

[31] P. Velickovic, D. Wang, N. D. Lane, and P. Lio, ''X-CNN: Cross-modal convolutional neural networks for sparse datasets,'' in Proc. IEEE Symp. Ser. Comput. Intell. (SSCI), Dec. 2016, pp. 1–8.

[32] Y.-H. H. Tsai, S. Bai, P. P. Liang, J. Z. Kolter, L.-P. Morency, and R. Salakhutdinov, ''Multimodal transformer for unaligned multimodal language sequences,'' in Proc. 57th Annu. Meeting Assoc. Comput. Linguistics, Jul. 2019, pp. 6558–6569. [Online]. Available: https://aclanthology.org/P19-1656

[33] C. E. Shannon, ''A mathematical theory of communication,'' Bell Syst. Tech. J., vol. 27, no. 3, pp. 379–423, 1948. [Online]. Available: http://plan9.belllabs.com/cm/ms/what/shannonday/shannon1948.pdf

[34] J. Kim, E.-S. Cho, and J.-Y. Paik, ''Poster: Feature engineering using file layout for malware detection,'' in Proc. Annu. Comput. Secur. Appl. Conf., Dec. 2020.

[35] M.-T. Luong, H. Pham, and C. D. Manning, ''Effective approaches to attention-based neural machine translation,'' in Proc. EMNLP, Aug. 2015, pp. 1412–1421. [Online]. Available: https://aclanthology.org/D15-1166

[36] J. Yan, G. Yan, and D. Jin, ''Classifying malware represented as control flow graphs using deep graph convolutional neural network,'' in Proc. 49th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN), Jun. 2019, pp. 52–63.

[37] M. Mays, N. Drabinsky, and S. Brandle, ''Feature selection for malware classification,'' in Proc. MAICS, Apr. 2017, pp. 165–170.

[38] Y. Zhang, Q. Huang, X. Ma, Z. Yang, and J. Jiang, ''Using multi-features and ensemble learning method for imbalanced malware classification,'' in Proc. IEEE Trustcom/BigDataSE/ISPA, Aug. 2016, pp. 965–973.

[39] M. Ahmadi, D. Ulyanov, S. Semenov, M. Trofimov, and G. Giacinto, ''Novel feature extraction, selection and fusion for effective malware family classification,'' in Proc. 6th ACM Conf. Data Appl. Secur. Privacy, Mar. 2016, pp. 183–194.

[40] R. Mitsuhashi and T. Shinagawa, ''Deriving optimal deep learning models for image-based malware classification,'' in Proc. 37th ACM/SIGAPP Symp. Appl. Comput. New York, NY, USA: Association for Computing Machinery, Apr. 2022, pp. 1727–1731, doi: 10.1145/3477314.3507242.

[41] J. H. Go, T. Jan, M. Mohanty, O. P. Patel, D. Puthal, and M. Prasad, ''Visualization approach for malware classification with ResNeXt,'' in Proc. IEEE Congr. Evol. Comput. (CEC), Jul. 2020, pp. 1–7.

[42] Y.-S. Liu, Y.-K. Lai, Z.-H. Wang, and H.-B. Yan, ‘‘A new learning approach to malware classification using discriminative feature extraction,’’ IEEE Access, vol. 7, pp. 13015–13023, 2019.

[43] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, and S. Venkatraman, ‘‘Robust intelligent malware detection using deep learning,’’ IEEE Access, vol. 7, pp. 46717–46738, 2019.

AUTHOR PROFILE

Mr. B. SURESH REDDY**,** done his M. Tech (Masters of Technology) in Arjun College of Technology & Sciences. At JNTU Hyderabad. Assistant Professor in the department of CSE at QIS College of Technology(Autonomous),Vengamukkapalem(V), Ongole, Prakasam. His areas of interest are Data Structures, Machine learning, and Web technologies.

Mr.VENKATA GOPALA KRISHNA.Y, currently pursuing Master of Computer Applications at QIS College of engineering and Technology (Autonomous), Ongole, Andhra Pradesh. He Completed B.SC(Computer scince) from SRI NIKHILA Degree College, chilakaloorpet, Andhra Pradesh. His areas of interests are Block chain, Artificial Intelligence, Data Science.