International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

A Robust Deep Learning Framework for Intelligent Malware Detection

S.Srikar[1], G.Rajini[2]
[1]MCA Student , Dept of MCA , Ellenki College of Engineering and Technology , Hyderabad
[2]Assistant Professor , Dept of MCA , Ellenki College of Engineering and Technology , Hyderabad

***Abstract*— Malware has become one of the biggest threats to computer systems and digital networks, affecting individuals, businesses, and government organizations. Traditional malware detection methods mainly depend on signatures and predefined rules, making them less effective against newly developed and constantly evolving attacks. As cybercriminals continue to use advanced techniques such as code obfuscation and polymorphism, there is a growing need for smarter and faster detection methods. This project proposes a robust malware detection system using deep learning to identify both known and unknown malware with improved accuracy. The model automatically learns meaningful patterns from malware data without relying heavily on manual feature extraction. By analyzing the behavior and characteristics of malicious files, the proposed system can classify malware efficiently while reducing false alarms. The approach improves detection speed, enhances security, and supports real-time threat analysis. This makes it a reliable solution for protecting modern computer systems against rapidly changing cyber threats.**

***Keywords* — Malware Detection, Deep Learning, Cybersecurity, Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), Machine Learning, Malware Classification, Zero-Day Malware, Threat Detection, Network Security.**

I. INTRODUCTION

The rapid growth of digital technologies, cloud computing, the Internet of Things (IoT), and online services has significantly increased the number of cyber threats targeting individuals, organizations, and government sectors. Among these threats, malware remains one of the most dangerous forms of cyberattacks because it can steal sensitive information, damage computer systems, encrypt valuable data, and disrupt critical services. Modern malware has evolved from simple viruses to sophisticated ransomware, spyware, trojans, worms, and botnets that continuously change their behavior to avoid detection. As cybercrime continues to increase, the financial and operational impact on organizations has become a serious concern worldwide [1].

Traditional malware detection techniques mainly rely on signature-based and rule-based approaches. Although these methods are effective in detecting previously known malware, they struggle to identify newly emerging and zero-day attacks. Malware developers frequently use advanced obfuscation techniques such as polymorphism, metamorphism, and code packing to generate multiple variants of the same malware, making signature-based detection ineffective [3], [5], [6]. Furthermore, static analysis cannot always identify malicious behavior, while dynamic analysis requires considerable execution time and computational resources, limiting its applicability in real-time environments [27].

To overcome these limitations, researchers have increasingly adopted machine learning techniques for malware detection. Machine learning algorithms such as Support Vector Machine (SVM), Decision Tree, Random Forest, Logistic Regression, and K-Nearest Neighbors have shown promising performance by learning patterns from malware features instead of relying solely on predefined signatures [8]. Public datasets such as EMBER have also enabled researchers to develop and evaluate intelligent malware detection systems under standardized conditions [26]. However, conventional machine learning methods often depend on manual feature engineering, which requires extensive domain expertise and may fail to capture complex malware characteristics.

Recent advancements in deep learning have opened new opportunities for intelligent malware detection. Deep learning models automatically learn hierarchical feature representations directly from raw executable files, API call sequences, process behavior, binary images, and network activities without extensive manual feature extraction [16]. Convolutional Neural Networks (CNNs) have demonstrated excellent performance in learning visual patterns from malware images and binary representations [18], [19], [33], while Recurrent

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks effectively model sequential behavioral information such as API calls and execution traces for malware classification [13], [20], [22], [24]. These models significantly improve detection accuracy while reducing false positives compared to traditional approaches.

Several recent studies have also explored end-to-end deep learning architectures that analyze complete executable files without requiring handcrafted features [11], [12]. Additionally, hybrid approaches combining static, dynamic, and behavioral analysis have further enhanced malware detection performance and improved robustness against sophisticated cyber threats [21], [23], [27]. Despite these advancements, challenges such as adversarial attacks, evolving malware variants, and large-scale data processing continue to motivate the development of more intelligent and adaptive detection frameworks [14], [15].

In this work, a Robust Intelligent Malware Detection Using Deep Learning framework is proposed to accurately classify malware using advanced deep learning techniques. The proposed system automatically extracts meaningful features from malware data, enabling effective identification of both known and previously unseen malware families. By leveraging deep neural networks, the system aims to improve classification accuracy, reduce detection time, and provide a scalable solution for real-time cybersecurity applications. The proposed approach contributes to strengthening modern cyber defense mechanisms by offering an efficient, reliable, and intelligent malware detection system capable of addressing the continuously evolving landscape of cyber threats.

II. RELATED WORK

"Large-Scale Identification of Malicious Singleton Files," B. Li, K. Roundy, C. Gates, and Y. Vorobeychik [2]
This study focused on identifying malicious singleton files that appear on only a single computer, making them difficult to detect using conventional signature-based techniques. The authors analyzed a large dataset containing billions of executable files collected from millions of systems and observed that most singleton files were benign, while a small portion represented newly emerging malware. A machine learning classifier based on static features was developed to distinguish malicious files from legitimate ones with high detection accuracy and a low false positive rate. The research demonstrated that static feature analysis combined with intelligent learning algorithms can effectively detect previously unseen malware without requiring code execution. The proposed approach also showed robustness against obfuscation and packing techniques commonly used by attackers. This work highlighted the importance of intelligent feature learning for scalable malware detection and motivated the development of advanced deep learning models capable of identifying zero-day malware efficiently.

"Zero-day Malware Detection Based on Supervised Learning Algorithms of API Call Signatures," M. Alazab, S. Venkatraman, P. Watters, and M. Alazab [5]
This research presented a supervised machine learning approach for detecting zero-day malware using Windows API call signatures. The authors collected a large dataset of malicious and benign programs and extracted API call frequencies to represent malware behavior. Several classification algorithms, including Naïve Bayes, k-Nearest Neighbor, Decision Tree, and Support Vector Machine, were evaluated to identify unknown malware samples. The experimental results demonstrated that behavioral analysis through API calls provides better generalization than traditional signature-based detection methods. The study emphasized that malware variants often share similar behavioral characteristics even when their code structure changes due to obfuscation techniques. This finding encouraged the use of intelligent learning algorithms for malware analysis instead of relying solely on predefined signatures. The research laid an important foundation for later deep learning approaches that automatically learn behavioral representations and improve malware detection accuracy against evolving cyber threats.

"Deep Neural Network Based Malware Detection Using Two Dimensional Binary Program Features," J. Saxe and K. Berlin [19]
This study introduced a deep neural network framework for malware detection by transforming executable files into two-dimensional binary feature representations. Instead of relying on manually engineered features, the proposed approach enabled the neural network to automatically learn complex malware characteristics directly from binary data. The model successfully captured hidden relationships among binary patterns, leading to improved classification accuracy compared to conventional machine learning techniques. The authors demonstrated that deep learning models are capable of identifying sophisticated malware variants while maintaining scalability for large datasets. The research also highlighted the effectiveness of neural networks in reducing feature engineering efforts and adapting to newly emerging malware families. The success of this work inspired

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

further research on applying convolutional neural networks and other deep learning architectures for intelligent malware detection, making it a significant contribution to modern AI-based cybersecurity solutions.

"Malware Detection with Deep Neural Network Using Process Behavior," S. Tobiyama, Y. Yamaguchi, H. Shimada, T. Ikuse, and T. Yagi [20]

This paper proposed a malware detection method based on process behavior using deep neural networks. Rather than analyzing only static executable features, the authors monitored the runtime behavior of applications and converted process activities into meaningful sequential representations. A deep learning model was then trained to distinguish malicious programs from legitimate software by learning behavioral patterns automatically. The proposed system demonstrated higher detection capability for unknown malware because behavioral characteristics are more difficult for attackers to disguise completely. Experimental results showed that deep neural networks effectively captured complex runtime relationships and significantly improved malware detection performance while maintaining low false alarm rates. The research emphasized the advantages of behavioral analysis over traditional signature matching and demonstrated the potential of deep learning for developing intelligent cybersecurity systems capable of detecting sophisticated and evolving malware attacks.

"Malicious Software Classification Using VGG16 Deep Neural Network's Bottleneck Features," E. Rezende, G. Ruppert, T. Carvalho, A. Theophilo, F. Ramos, and P. de Geus [18]

This study investigated the application of the VGG16 deep convolutional neural network for malware classification using bottleneck features extracted from malware images. The authors converted executable files into grayscale images and utilized transfer learning to obtain rich visual representations without training a deep network from scratch. These extracted features were then used for accurate malware family classification. Experimental results demonstrated that the proposed method achieved high classification accuracy while reducing computational complexity and training time. The research proved that malware images contain distinctive visual patterns that deep convolutional networks can effectively recognize. The use of transfer learning also improved model generalization and minimized the need for extensive labeled datasets. This work demonstrated the practical advantages of deep learning in cybersecurity and inspired many subsequent studies on image-based malware detection using convolutional neural network architectures.

III. DATASET DETAILS

The dataset used in this study consists of Portable Executable (PE) files collected from both benign software and malicious applications. It includes samples representing multiple malware families as well as legitimate executable files, enabling the development of a supervised malware classification system. Each executable contains valuable information such as PE header attributes, imported libraries, section information, file size, entropy, API calls, and other static features that characterize the behavior of the program. The dataset contains labeled records, where each sample is categorized as either Benign or Malware, with malware further belonging to different malware families. Since the collected executable files may contain redundant, missing, or inconsistent feature values, preprocessing is required before model training. The dataset provides sufficient diversity to represent real-world malware behavior, allowing machine learning and deep learning models to learn meaningful patterns for identifying both known and previously unseen malware. By utilizing this dataset, the proposed system aims to develop an intelligent malware detection framework capable of accurately distinguishing malicious software from legitimate applications, thereby improving cybersecurity and reducing the risk of malware attacks.

Before training the proposed model, the dataset undergoes several preprocessing stages to improve its quality and enhance classification performance. Initially, missing values, duplicate records, and irrelevant attributes are removed to ensure data consistency. The extracted PE features are normalized and transformed into numerical representations suitable for machine learning and deep learning algorithms. Feature selection techniques are then applied to retain only the most informative attributes while reducing computational complexity. After preprocessing, the dataset is divided into training and testing subsets using an 80:20 ratio, where 80% of the samples are used for model training and the remaining 20% are reserved for performance evaluation. The processed dataset is then utilized by traditional machine learning algorithms such as Support Vector Machine (SVM), Random Forest (RF), and Decision Tree (DT), along with deep learning models such as Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM). This enables a comprehensive comparison of conventional and deep learning approaches, supporting the development of a robust

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

and intelligent malware detection system with high detection accuracy and low false positive rates.

IV. PROPOSED METHODOLOGY

The proposed methodology utilizes Portable Executable (PE) malware files to automatically identify and classify malicious software using both machine learning and deep learning techniques. Initially, the malware dataset is uploaded into the system, where records belonging to different malware families and benign executable files are collected for analysis. The uploaded dataset undergoes preprocessing to improve data quality by handling missing values, removing duplicate records, normalizing feature values, and eliminating inconsistencies that may affect classification performance. After preprocessing, feature extraction techniques are applied to obtain meaningful characteristics from the executable files, including PE header information, binary features, and other static malware attributes. These extracted features represent important behavioral and structural patterns associated with malicious software. The processed dataset is then divided into training and testing sets using an 80:20 ratio. This preparation stage ensures that the classification models receive reliable and informative data for learning and accurate malware detection.

Following data preparation, the extracted malware features are provided to both the existing machine learning algorithms and the proposed deep learning models for training and classification. Initially, Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Naïve Bayes (NB), Decision Tree (DT), Logistic Regression (LR), and Random Forest (RF) algorithms are trained using the prepared dataset to classify malware samples. Subsequently, the proposed MalConv Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) models automatically learn complex malware patterns through multiple convolution, sequential learning, and feature extraction layers without requiring extensive manual feature engineering. During training, the deep learning models continuously update their parameters over multiple epochs to improve their ability to distinguish between benign applications and different malware families. After training, the models are evaluated using the testing dataset to measure their classification effectiveness. Performance metrics such as Accuracy, Precision, Recall, and F1-Score are calculated to assess prediction quality. The trained deep learning models are further used to predict the malware family of unseen executable files. Finally, the obtained results are compared with the existing machine learning algorithms using graphical and statistical performance analysis.

The system architecture [1] illustrates the complete workflow of the proposed deep learning-based malware detection system. It includes malware dataset uploading, preprocessing, feature extraction, SVM, KNN, Naïve Bayes, Decision Tree, Logistic Regression, Random Forest, MalConv CNN, and LSTM model training, malware family prediction from unknown executable files, and performance evaluation. The architecture also generates confusion matrices, detailed performance metrics, malware prediction history, and comparison graphs to analyze the effectiveness of all classification algorithms.

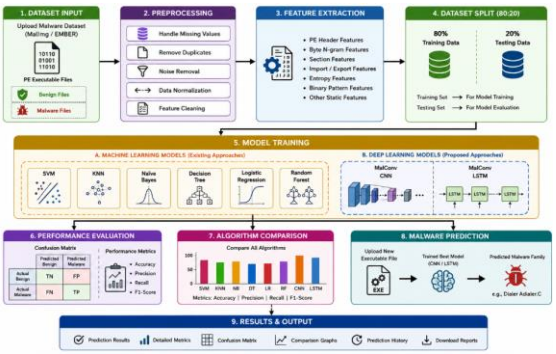

Figure 1. System Architecture for Robust Intelligent Malware Detection Using Deep Learning

The proposed methodology for malware detection is designed to accurately identify malicious executable files using machine learning and deep learning techniques. First, the malware dataset is uploaded into the system, and executable files are separated into benign and malware categories representing multiple malware families. Since the collected dataset may contain missing values, duplicate records, or irrelevant attributes, preprocessing is performed to improve data quality. After cleaning the dataset, important static malware features are extracted from the executable files to represent their structural and behavioral characteristics in numerical form. These extracted features enable the classification models to effectively distinguish between legitimate applications and malicious software.

After feature extraction, the dataset is divided into training and testing data using an 80:20 ratio. The training dataset is used to train both the existing machine learning algorithms (SVM, KNN, Naïve

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

Bayes, Decision Tree, Logistic Regression, and Random Forest) and the proposed deep learning models (MalConv CNN and LSTM), while the testing dataset is used to evaluate their performance. The machine learning algorithms serve as the existing approaches, whereas MalConv CNN and LSTM are employed as the proposed deep learning models because they automatically learn complex malware representations directly from executable files. Finally, the trained models predict the malware family of newly uploaded binary files. The performance of all algorithms is compared using Accuracy, Precision, Recall, F1-Score, confusion matrices, detailed metrics, and comparison graphs, demonstrating the effectiveness and robustness of the proposed intelligent malware detection framework.

V. RESULT AND DISCUSSION

The experimental results demonstrate the effectiveness of the proposed MalConv Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) models for intelligent malware detection and malware family classification. Initially, the MalImg malware dataset was uploaded into the system and preprocessed to extract meaningful malware features from executable files. The processed dataset was divided into training and testing subsets using an 80:20 ratio, ensuring a fair evaluation of the classification models. Existing machine learning algorithms including Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Naïve Bayes, Decision Tree, Logistic Regression, and Random Forest were executed to establish baseline performance. Among these classifiers, the SVM algorithm produced satisfactory classification results with an accuracy of 86.88%, while the remaining algorithms also generated competitive performance. However, traditional machine learning methods rely heavily on manually extracted features and have limited capability in learning complex malware characteristics automatically.

Subsequently, the proposed MalConv CNN and LSTM models were trained using the same dataset. The CNN model automatically learned discriminative malware features directly from executable files through multiple convolutional layers, whereas the LSTM model captured sequential malware characteristics for classification. Experimental observations showed that the proposed CNN model achieved an accuracy of 87.50%, slightly outperforming the existing SVM model. The LSTM model obtained an accuracy of 83.37%, demonstrating competitive performance while effectively learning malware behavior patterns. Comparison graphs generated by the system clearly illustrated the performance differences among all algorithms using Accuracy, Precision, Recall, and F1-Score metrics. Furthermore, the malware prediction module successfully classified unseen executable files and correctly identified their malware family. The prediction history maintained by the system enables users to review previous malware classifications for further analysis. Overall, the experimental findings demonstrate that deep learning models provide an intelligent and automated solution for malware detection and family classification in modern cybersecurity environments.

Figure [2] Malware Dataset Successfully Uploaded

Figure 2 shows the successful loading of the MalImg malware dataset into the proposed system. After selecting the dataset, the application confirms that the dataset has been loaded successfully and is ready for preprocessing and model training. This stage initializes the complete malware detection workflow and prepares the dataset for feature extraction and classification.

Figure 3: Performance of Existing SVM Classifier

Figure 3 presents the prediction results generated by the existing Support Vector Machine (SVM) classifier. The model achieved an Accuracy of 86.875%, Precision of 86.875%, Recall of 86.875%, and F1-Score of 86.875%. These results indicate that the SVM algorithm performs well in malware classification but has limitations in automatically

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

learning complex malware patterns compared with deep learning approaches.

Figure 4: Performance of Proposed MalConv CNN

Figure 4 illustrates the prediction results of the proposed MalConv CNN model. The CNN classifier achieved an Accuracy of 87.50%, Precision of 87.50%, Recall of 87.50%, and F1-Score of 87.50%. These results demonstrate that the CNN model provides improved malware classification performance by automatically extracting hierarchical malware features directly from executable files.

Figure 5: Performance of Proposed LSTM Model

Figure 5 presents the prediction results of the proposed LSTM classifier. The model achieved an Accuracy of 83.37%, Precision of 83.37%, Recall of 83.37%, and F1-Score of 83.37%. Although its performance is slightly lower than the CNN model, the LSTM effectively captures sequential malware characteristics and contributes to intelligent malware detection.

Model	Accuracy score	Precision score	Recall score	f-measure score
SVM	86.88	86.88	86.88	86.88
KNN	85.42	85.42	85.42	85.42

Model	Accuracy score	Precision score	Recall score	f-measure score
CNN	87.50	87.50	87.50	87.50
LSTM	83.37	83.37	83.37	83.37

Table 1: Model Performance Metrics

Figure 6: Malware Family Prediction

Figure 7 illustrates the malware family prediction module of the proposed system. After uploading an unknown executable file (1.npy), the trained deep learning model successfully predicts that the uploaded sample belongs to the Dialer.Adialer.C malware family. This demonstrates the capability of the proposed framework to accurately classify previously unseen malware samples and identify their corresponding malware families.

DISCUSSION

The experimental findings demonstrate the effectiveness of machine learning and deep learning techniques for automated malware detection and malware family classification. The MalImg dataset contained executable files belonging to multiple malware families, enabling the classification models to learn meaningful malware characteristics. Existing machine learning algorithms such as SVM, KNN, Naïve Bayes, Decision Tree, Logistic Regression, and Random Forest produced satisfactory performance; however, they rely on manually engineered features and therefore exhibit limited capability in capturing complex malware representations. Among the existing approaches, the SVM classifier achieved an accuracy of 86.88%, indicating good malware detection capability.

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

The proposed MalConv CNN model achieved the highest classification accuracy (87.50%) by automatically learning hierarchical malware features directly from executable files without extensive manual feature engineering. The LSTM model also demonstrated competitive performance with an accuracy of 83.37%, effectively modeling sequential malware characteristics. The comparison of evaluation metrics confirmed the superiority of the CNN model over the other classifiers. In addition, the malware prediction module successfully identified unseen executable files and correctly classified the uploaded sample as Dialer.Adialer.C, demonstrating the model's ability to generalize to new malware samples. Overall, the proposed deep learning-based malware detection framework provides an efficient, reliable, and automated solution for intelligent malware detection and malware family classification, making it suitable for strengthening modern cybersecurity systems against evolving malware threats.

VI. CONCLUSION

This work presented a Robust Intelligent Malware Detection Using Deep Learning framework for accurately detecting and classifying malicious executable files. The proposed system integrates both traditional machine learning algorithms and deep learning models to analyze malware samples collected from the MalImg dataset. Initially, the malware dataset undergoes preprocessing and feature extraction to obtain meaningful representations of executable files. Existing machine learning algorithms, including Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Naïve Bayes (NB), Decision Tree (DT), Logistic Regression (LR), and Random Forest (RF), were evaluated and compared with the proposed MalConv Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) models. Experimental results demonstrated that deep learning techniques provide improved malware detection capability by automatically learning complex malware characteristics without relying heavily on manual feature engineering.

The proposed MalConv CNN achieved the highest classification performance among all evaluated models, while the LSTM model also demonstrated competitive results by effectively capturing sequential malware patterns. The developed framework successfully identifies malware families from unknown executable files and provides detailed performance evaluation through accuracy, precision, recall, F1-score, comparison graphs, and malware prediction reports. The graphical user interface further simplifies malware analysis by enabling dataset uploading, model execution, performance visualization, and malware family prediction within a single platform.

Although the proposed framework provides reliable malware detection, there is still scope for improvement. Future work can incorporate larger and more diverse malware datasets, memory dump analysis, dynamic behavioral features, attention-based deep learning architectures, and explainable artificial intelligence (XAI) techniques to improve detection accuracy and model interpretability. Additionally, integrating real-time cloud-based malware analysis and federated learning can enhance scalability and enable efficient detection of emerging zero-day malware threats. Overall, the proposed deep learning-based malware detection framework offers an efficient, scalable, and intelligent solution for strengthening modern cybersecurity systems against continuously evolving malware attacks.

REFERENCES

[1] Anderson, R., Barton, C., Böhme, R., Clayton, R., Van Eeten, M. J., Levi, M., ... & Savage, S. (2013). Measuring the cost of cybercrime. In The economics of information security and privacy (pp. 265-300). Springer, Berlin, Heidelberg.

[2] Li, B., Roundy, K., Gates, C., & Vorobeychik, Y. (2017, March). LargeScale Identification of Malicious Singleton Files. In Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy (pp. 227-238). ACM.

[3] Alazab, M., Venkataraman, S., & Watters, P. (2010, July). Towards understanding malware behaviour by the extraction of API calls. In 2010 Second Cybercrime and Trustworthy Computing Workshop (pp. 52-59). IEEE.

[4] Tang, M., Alazab, M., & Luo, Y. (2017). Big data for cybersecurity: vulnerability disclosure trends and dependencies. IEEE Transactions on Big Data.

[5] Alazab, M., Venkatraman, S., Watters, P., & Alazab, M. (2011, December). Zero-day malware detection based on supervised learning algorithms of API call signatures. In Proceedings of the Ninth Australasian Data Mining Conference-Volume 121 (pp. 171-182). Australian Computer Society, Inc..

[6] Alazab, M., Venkatraman, S., Watters, P., Alazab, M., & Alazab, A. (2011, January).

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

Cybercrime: the case of obfuscated malware. In 7th ICGS3/4th e-Democracy Joint Conferences 2011: Proceedings of the International Conference in Global Security, Safety and Sustainability/International Conference on e-Democracy (pp. 1-8). [Springer].

[7] Alazab, M. (2015). Profiling and classifying the behavior of malicious codes. Journal of Systems and Software, 100, 91-102.

[8] Huda, S., Abawajy, J., Alazab, M., Abdollalihian, M., Islam, R., & Yearwood, J. (2016). Hybrids of support vector machine wrapper and filter based framework for malware detection. Future Generation Computer Systems, 55, 376-390.

[9] Raff, E., Sylvester, J., & Nicholas, C. (2017, November). Learning the PE Header, Malware Detection with Minimal Domain Knowledge. In Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security (pp. 121-132). ACM.

[10] Rossow, C., Dietrich, C. J., Grier, C., Kreibich, C., Paxson, V., Pohlmann, N., ... & Van Steen, M. (2012, May). Prudent practices for designing malware experiments: Status quo and outlook. In Security and Privacy (SP), 2012 IEEE Symposium on (pp. 65-79). IEEE.

[11] Raff, E., Barker, J., Sylvester, J., Brandon, R., Catanzaro, B., & Nicholas, C. (2017). Malware detection by eating a whole exe. arXiv preprint arXiv:1710.09435.

[12] Krcál, M., ˇ Svec, O., Bálek, M., & Ja ˇ sek, O. (2018). Deep Convolutional ˇ Malware Classifiers Can Learn from Raw Executables and Labels Only.

[13] Rhode, M., Burnap, P., & Jones, K. (2018). Early-stage malware prediction using recurrent neural networks. Computers & Security, 77, 578-594.

[14] Anderson, H. S., Kharkar, A., Filar, B., & Roth, P. (2017). Evading machine learning malware detection. Black Hat.

[15] Verma, R. (2018, March). Security Analytics: Adapting Data Science for Security Challenges. In Proceedings of the Fourth ACM International Workshop on Security and Privacy Analytics (pp. 40-41). ACM.

[16] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436.

[17] Agarap, A. F., & Pepito, F. J. H. (2017). Towards Building an Intelligent Anti-Malware System: A Deep Learning Approach using Support Vector Machine (SVM) for Malware Classification. arXiv preprint arXiv:1801.00318.

[18] Rezende, E., Ruppert, G., Carvalho, T., Theophilo, A., Ramos, F., & de Geus, P. (2018). Malicious Software Classification Using VGG16 Deep Neural Network's Bottleneck Features. In Information Technology-New Generations (pp. 51-59). Springer, Cham.

[19] Saxe, J., & Berlin, K. (2015, October). Deep neural network based malware detection using two dimensional binary program features. In Malicious and Unwanted Software (MALWARE), 2015 10th International Conference on (pp. 11-20). IEEE.

[20] Tobiyama, S., Yamaguchi, Y., Shimada, H., Ikuse, T., & Yagi, T. (2016, June). Malware detection with deep neural network using process behavior. In Computer Software and Applications Conference (COMPSAC), 2016 IEEE 40th Annual (Vol. 2, pp. 577-582). IEEE.

[21] Huang, W., & Stokes, J. W. (2016, July). MtNet: a multi-task neural network for dynamic malware classification. In International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (pp. 399-418). Springer, Cham.

[22] Pascanu, R., Stokes, J. W., Sanossian, H., Marinescu, M., & Thomas, A. (2015, April). Malware classification with recurrent networks. In Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on (pp. 1916-1920). IEEE.

[23] Shibahara, T., Yagi, T., Akiyama, M., Chiba, D., & Yada, T. (2016, December). Efficient dynamic malware analysis based on network behavior using deep learning. In Global Communications Conference (GLOBECOM), 2016 IEEE (pp. 1-7). IEEE.

[24] Kolosnjaji, B., Zarras, A., Webster, G., & Eckert, C. (2016, December). Deep learning for classification of malware system call sequences. In Australasian Joint Conference on Artificial Intelligence (pp. 137-149). Springer, Cham.

[25] Raff, E., Zak, R., Cox, R., Sylvester, J., Yacci, P., Ward, R., ... & Nicholas, C. (2018). An investigation of byte n-gram features for malware

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

classification. Journal of Computer Virology and Hacking Techniques, 14(1), 1-20.

[26] Anderson, H. S., & Roth, P. (2018). EMBER: An Open Dataset for Training Static PE Malware Machine Learning Models. arXiv preprint arXiv:1804.04637.

[27] Damodaran, A., Di Troia, F., Visaggio, C. A., Austin, T. H., & Stamp, M. (2017). A comparison of static, dynamic, and hybrid analysis for malware detection. Journal of Computer Virology and Hacking Techniques, 13(1), 1-12. [28] Nataraj, L. (2015). A signal processing approach to malware analysis. University of California, Santa Barbara.

[29] Nataraj, L., Kirat, D., Manjunath, B. S., & Vigna, G. (2013, December). Sarvam: Search and retrieval of malware. In Proceedings of the Annual Computer Security Conference (ACSAC) Worshop on Next Generation Malware Attacks and Defense (NGMAD).

[30] Nataraj, L., Yegneswaran, V., Porras, P., & Zhang, J. (2011, October). A comparative assessment of malware classification using binary texture analysis and dynamic analysis. In Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence (pp. 21-30). ACM.

[31] Nataraj, L., Jacob, G., & Manjunath, B. S. (2010). Detecting packed executables based on raw binary data. Technical report.

[32] Farrokhmanesh, M., & Hamzeh, A. (2016, April). A novel method for malware detection using audio signal processing techniques. In Artificial Intelligence and Robotics (IRANOPEN), 2016 (pp. 85-91). IEEE.

[33] Nataraj, L., Karthikeyan, S., Jacob, G., & Manjunath, B. S. (2011, July). Malware images: visualization and automatic classification. In Proceedings of the 8th international symposium on visualization for cyber security (p. 4). ACM.

[34] Nataraj, L., & Manjunath, B. S. (2016). SPAM: signal processing to analyze malware. arXiv preprint arXiv:1605.05280.

[35] Kirat, D., Nataraj, L., Vigna, G., & Manjunath, B. S. (2013, December). Sigmal: A static signal processing based malware triage. In Proceedings of the 29th Annual Computer Security Applications Conference (pp. 89-98). ACM.

[36] Akinapalli, S. Agentic AI for Autonomous Data Engineering Operations in Cloud-Native Data Ecosystems.Journal of Information Systems Engineering and Management, 2024, 9(2)