INSTITUTE FOR HOMELAND SECURITY

Sam Houston State University

Leveraging Large Language Models for Behavior-Based Malware Detection Using Deep Learning

Institute for Homeland Security

Sam Houston State University

Tosin Akinsowon

Haodi Jiang

Abstract

Abstract – Malware poses significant threats through cybercrime, fraud, scams, and nation-state cyberwarfare, often resulting in irreversible data loss and substantial economic damages. Traditional antivirus solutions, relying on signature-based and heuristic-based detections, struggle against zero-day attacks. Heuristic methods analyze the behavior of programs to detect potentially malicious patterns, yet they still fall short when facing sophisticated malware. Furthermore, the adoption of Large Language Models (LLMs) such as GPT-4 and LLaMA by cyber attackers complicates the identification of malicious activities, increasing both alert fatigue and investigative costs. This proposal introduces a deep learning framework designed to overcome these limitations by utilizing artificial intelligence to analyze malware based on dynamic behavioral features captured in a controlled sandbox environment. The framework employs LLMs to abstract these behavioral features, thereby enhancing detection capabilities. This innovative approach not only promises to improve malware detection rates but also sets the stage for future advancements in cybersecurity technologies.

Index Terms – Malware Dynamic Behavior, Malware Detection, Large Language Models, Machine Learning

I. Introduction and Overview

The relentless evolution of malware poses significant challenges to cybersecurity, necessitating innovative solutions for the timely and accurate detection and classification of these threats. Malware, a severe risk to computer systems, exploits vulnerabilities to gain unauthorized access and cause damage. Traditional signature-based detection methods often fall short in addressing the rapidly mutating and obfuscating nature of modern malware. Consequently, there is growing interest in exploring deep learning approaches for more effective malware analysis and classification [1]. Given the success of deep learning in various tasks such as recognition, summarization, translation, prediction, and content generation, these methods are increasingly being proposed for malware classification [2]. They possess the capability to identify complex patterns and extract meaningful representations directly from raw data.

Conversely, researchers have thoroughly explored the application of Large Language Models (LLMs) for text embedding across various domains. For example, Li and Yu investigated the effectiveness of LLMs like GPT-2 in generating text embeddings, demonstrating that these models can maintain semantic integrity while greatly enhancing computational efficiency [3]. Similarly, Cao and Gao employed LLMs for malware traffic classification, showing that LLM-generated embeddings effectively differentiate between benign and malicious traffic, resulting in faster processing times and improved accuracy [4]. In another study, Singh and Sharma examined the impact of LLM-generated text embeddings, emphasizing how LLMs preserve essential features required for high performance in natural language processing tasks [5].

More recently, researchers have focused on leveraging advanced language models and dynamic behavior analysis for malware detection, driven by the need to better understand and classify the increasingly complex and sophisticated nature of malicious software. Large Language Models (LLMs) were utilized for behavior analysis in malware detection due to their ability to process and interpret vast amounts of unstructured data, extract meaningful patterns, and identify subtle indicators of malicious activities that traditional methods might overlook. Li and Yu examined the application of LLMs in analyzing malware behavior, showcasing their effectiveness in identifying and understanding these malicious activities [3]. Chen et al. utilized LLMs to process and analyze report.json files from Cuckoo Sandbox, introducing a novel approach for dynamic malware detection [6]. Wang and Li proposed a deep learning-based method that combines dynamic behavior analysis with report.json files from Cuckoo Sandbox, demonstrating the effectiveness of deep learning techniques, including LLMs, in enhancing the accuracy and robustness of malware detection [7]. These studies highlight the significant potential of LLMs and dynamic behavior analysis in advancing malware detection tasks.

II. Gap Assessment and Problem Statement

The increasing complexity and sophistication of modern malware has surpassed the capabilities of traditional detection methods, making them insufficient for addressing current threats. Even with technological advancements, existing machine learning (ML) and deep learning (DL) techniques continue to face challenges due to the diverse and rapidly evolving nature of malware. While LLMs show promise in analyzing complex malware behaviors, their computational efficiency and ability to capture subtle behaviors still require refinement. Additionally, dynamic behavior analysis offers deeper insights into malware activities, but converting these behaviors into actionable features for ML/DL models remains a significant challenge. Furthermore, the continuous need to manage large, high-quality datasets that accurately reflect the evolving threat landscape adds another layer of complexity to the task.

The increasing complexity of malware necessitates advanced detection methods that surpass traditional techniques. While machine learning and deep learning have shown promise, they often require significant feature engineering and face challenges with generalization. LLMs into dynamic behavior analysis opens up new possibilities, but further research is needed to improve computational efficiency and accuracy. The key challenge is to develop an

innovative deep learning framework that combines LLMs with dynamic behavior analysis for malware classification, aiming to enhance detection accuracy, robustness, and efficiency in real-world applications.

This research is strategically positioned to address critical cybersecurity threats, particularly benefiting multiple critical infrastructure sectors. The primary objective is to create an innovative Artificial Intelligence (AI) tool designed for malware detection and analysis. This tool will serve as a preemptive system to enhance the cybersecurity of computer systems, especially within hospitals, medical centers, and the transportation department across Texas. By directly tackling the critical concept of cybersecurity threats, this research will make significant contributions to the field of cybersecurity and offer protection to essential infrastructure sectors.

III. Topic Discussion

In this section, we detail the dataset employed in our research, outlining its structure, the preparation process for malware classification, and the methodologies.

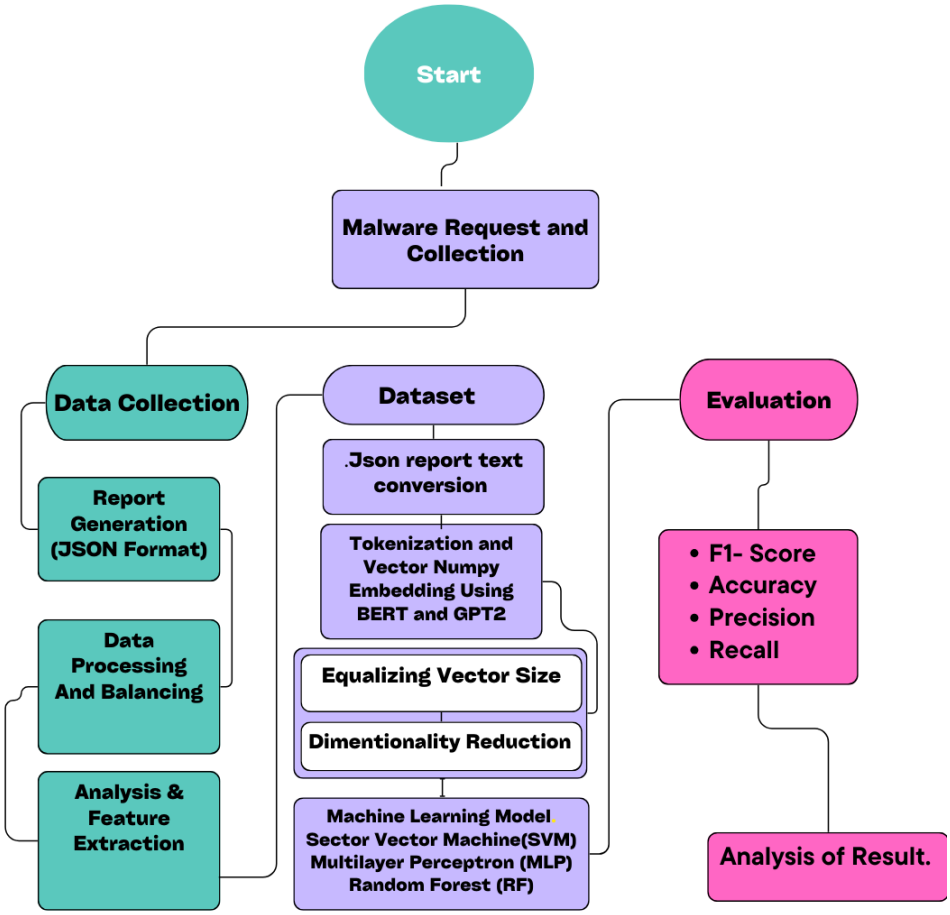

Fig.1 Overall Workflow

Figure 1 outlines the comprehensive workflow for performing multiclass classification on malware using dynamic behavior analysis and deep learning techniques with LLMs. The process is divided into ten phases. Initially, 48,976 samples from the CAPEV2 malware dataset are collected. During the dynamic behavior analysis phase, malware interactions are observed as they are originally captured in JSON reports. The data processing phase then prepares this data for further analysis. Key features are extracted from the dynamic behavior reports, providing insights into network activity, file operations, processes, and more. The dataset is refined to 20,000 samples, representing five families of malicious files while retaining the original JSON reports.

In the data transformation phase, the JSON files are converted to text and tokenized using LLMs, transforming the text data into numerical vectors. Vector size equalization ensures consistency in length through padding and truncation. Dimensionality reduction via Principal Component Analysis (PCA) standardizes the vectors to 10,000 dimensions. Machine learning models, including Support Vector Machine (SVM), Multilayer Perceptron (MLP), and Random Forest (RF), are then employed for classification. The performance of these models is evaluated using metrics such as F1-score, accuracy, precision, and recall. Finally, the results are analyzed to compare the models, identifying the best-performing one for malware classification.

A. Dataset Overview

A.1 Data Source

This study leverages the Avast-CTU Public CAPEv2 Dataset [8], accessible at https://github.com/avast/avast-ctu-cape-dataset. The dataset comprises logs collected from July to September 2021 on CAPEv2 instances, which were operated by the Artificial Intelligence Center at the Department of Computer Science, Faculty of Electrical Engineering, Czech Technical University. This initiative was conducted in collaboration with Avast Software, exemplifying the importance of industry-academia partnerships in addressing complex challenges in cybersecurity.

A.2 Data Composition

The CAPEv2 dataset initially includes detailed reports on approximately 49,000 malicious files, distributed across ten malware families: Adload, Emotet, HarHar, Lokibot, njRAT, Qakbot, Swisyn, Trickbot, Ursnif, and Zeus. It provides essential information for malware family classification, enabling the identification of patterns and similarities across various threats. The dataset categorizes samples into six malware types—banker, trojan, password stealer (pws), coin miner, remote access trojan (rat), and keylogger—and includes the date of detection, facilitating the tracking of malware evolution over time. For our analysis, we curated a balanced subset of 20,000 samples, evenly distributed across five selected families: Emotet, Lokibot, Qakbot, Swisyn, and Trickbot, with each family represented by 4,000 samples. The integrity and uniqueness of each file are ensured through the inclusion of SHA256 cryptographic hash values.

A.3 Data Preprocessing

To enable effective machine learning analysis, this study focuses on the dynamic behavior of malware as the basis for classification. Dynamic features such as executed files, network activity (including ports and IPs), and system modifications were extracted from the CAPEv2 reports. A sub-dataset of 20,000 SHA256 files was selected, concentrating on five malware families as mentioned previously. These JSON files, containing detailed behavioral data, were converted into plain text (.txt) format compatible with the LLMs API using Python scripts. With its diversity and scale, this dataset provides a robust foundation for developing machine learning models capable of detecting specific malware types and identifying generalized malicious behaviors that can adapt to evolving threats. Our research aims to advance the field of automatic malware detection by innovatively applying AI techniques, including LLMs and dynamic analysis, to enhance detection accuracy, robustness, and computational efficiency in real-world cybersecurity applications.

B. Malware Behavior Feature Vectorization Using LLMs

Text embedding is a key technique in natural language processing that transforms text data into numerical values or vectors, allowing machine learning models to process and understand the data more effectively. This study employs text embedding to represent the dynamic behavior of malware, using LLMs like BERT (Bidirectional Encoder Representations from Transformers) [9] to extract meaningful features from unstructured data. These features are crucial for identifying key behaviors and patterns that characterize different types of malware, and they are converted into numerical vectors that serve as input for machine learning models. Each vector element represents the presence, absence, or frequency of specific behaviors or attributes identified in the malware reports.

The advantage of using text embedding lies in its ability to capture the contextual meaning of words and phrases, which is essential for accurately understanding and classifying malware behaviors. By leveraging LLMs, we can process vast amounts of textual data from dynamic behavior reports, extracting intricate details that might be missed by traditional feature engineering methods. This approach not only enhances the accuracy of malware detection but also improves the robustness of the models against variations and obfuscations used by sophisticated malware. Additionally, text embeddings facilitate the integration of various data sources, enabling a more comprehensive analysis of malware activities. For instance, network logs, file operations, and system processes can all be embedded and analyzed within a unified framework, aiding in the identification of complex attack patterns and providing deeper insights into the malware's operational tactics.

In this study, we utilize the BERT model from the Hugging Face transformers library, specifically the "Bert base-uncased" model, which was pre-trained with a hidden size of 10,000. The text data—representations of malware files—are tokenized and encoded using the BERT tokenizer, which converts the text into tokens and generates the corresponding input IDs and attention masks. These are then fed into the BERT model to produce embeddings.

The embedding process involves passing the tokenized input through the BERT model to generate embeddings. The output, outputs[0], contains the hidden states from the last layer of the BERT model for each token in the input sequence. This output has a shape of (batch size, sequence length, hidden size), with each token represented by a 10,000-dimensional vector. To facilitate easier manipulation and storage, the batch dimension is removed using the pooled_output.squeeze(0).numpy() function, leaving a 2D array with the shape of (sequence length, 10,000). This array, which is then converted to a numpy array, collectively captures the meaning of the entire input text, with each vector representing the contextualized meaning of each token.

After generating these high-dimensional vectors, we face the challenge of non-uniform vector sizes due to varying file sizes. To achieve uniformity and reduce the dimensionality, we apply PCA using the sklearn library. The steps involved include setting up the environment, defining functions for loading vectors, applying PCA, and saving the reduced vectors along with the PCA model. The PCA model reduces the vectors to 10,000 dimensions, which balances the trade-off between capturing the most relevant information and ensuring computational efficiency.

This method aligns well with ongoing research efforts, demonstrating the potential for LLMs like BERT to enhance malware detection accuracy and robustness through dynamic behavior analysis. By continuously updating the models with new data and refining the embeddings, we can stay ahead of emerging threats and maintain a high level of security. The processed data, along with the embeddings, are saved in a .csv format for further analysis and model training. This approach is both cost-effective and capable of handling complex data, making it well-suited for real-world cybersecurity applications.

C. Malware Classification

To evaluate the effectiveness of our malware representations extracted using Large Language Models, we employed three traditional machine learning algorithms: Support Vector Machine (SVM) [10], Random Forest (RF) [13], and Multilayer Perceptron (MLP) [15]. These algorithms were chosen for their diverse approaches to handling high-dimensional data and their proven efficacy in various classification tasks, including malware detection.

SVM is a powerful classification algorithm that works by finding the optimal hyperplane that separates data points into different classes in a high-dimensional space. In the context of malware classification, SVM distinguishes between different malware categories by maximizing the margin between them. This margin maximization helps in creating a robust decision boundary that is less prone to misclassifications, especially in high-dimensional spaces where the number of dimensions often exceeds the number of samples. SVM is particularly effective for malware classification due to its ability to handle high-dimensional feature spaces and its robustness to overfitting, making it suitable for complex datasets [10].

Random Forest is an ensemble learning method that constructs a multitude of decision trees during training and outputs the class that is the mode of the classes predicted by the individual trees. In malware classification, RF operates by building trees based on the feature vectors

extracted from malware samples and aggregating their predictions to determine the most likely malware category. The ensemble approach of RF is advantageous as it reduces the risk of overfitting by averaging the results of multiple deep decision trees, leading to a more generalized model. RF is particularly adept at handling large datasets with high dimensionality, making it well-suited for complex malware classification tasks. Additionally, RF's ability to provide feature importance measures is valuable for understanding which features contribute most to the classification decisions [13].

MLPs are a class of feedforward artificial neural networks that consist of multiple layers of interconnected neurons. Each neuron processes input data through weighted connections and nonlinear activation functions, ultimately producing an output. In malware classification, MLPs analyze feature vectors derived from malware samples, enabling them to categorize the samples into distinct classes. During training, MLPs adjust their weights through backpropagation, a process that minimizes classification error and enhances model performance. MLPs are particularly effective in handling intricate, high-dimensional data, as they can extract nuanced patterns indicative of malicious activity. Their deep architecture, consisting of multiple hidden layers, helps mitigate overfitting, ensuring robust generalization to unseen samples. Additionally, MLPs are highly adaptable, capable of adjusting their weights in response to evolving threats, making them well-suited for dynamic environments where new malware variants frequently emerge. This adaptability and resilience make MLPs a robust framework for accurate and efficient malware classification [11] [12] [15].

We implemented the base models using the Python scikit-learn package, with specific hyperparameter settings to optimize performance. For the Support Vector Machine (SVM), we employed the Radial Basis Function (RBF) kernel. Given that our samples are high-dimensional and may not be linearly separable, the RBF kernel is particularly well-suited for capturing the complex, non-linear relationships within the data. This choice enables the SVM model to effectively differentiate between various malware categories by mapping the data into a higher-dimensional space where a clear separation is achievable. For the Random Forest model, we used an ensemble of 500 trees, with four randomly chosen features considered when searching for the best split at each node. This setup allows the RF model to balance complexity with the risk of overfitting, making it robust for handling large, high-dimensional datasets. The Multilayer Perceptron model was configured with three hidden layers, each consisting of 100 neurons, providing sufficient depth to capture intricate patterns in the data while maintaining computational efficiency. For any parameters not explicitly mentioned, default values were used.

This combination of traditional machine learning models, when applied to our LLM-extracted malware representations, provides a comprehensive evaluation of the effectiveness of our approach. Each model offers unique strengths in handling high-dimensional data, ensuring that our analysis covers various aspects of malware classification, from robustness and adaptability to accuracy and computational efficiency.

D. Performance Metrics

The performance of our models was evaluated using the following metrics:

1) *Accuracy:* Accuracy measures the proportion of correctly classified samples out of the total number of samples in the dataset. It is a commonly used metric in classification tasks, providing an overall indication of how well the model performs in distinguishing different types of malwares.

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

2) *Precision:* Precision quantifies the proportion of correctly predicted instances of a specific malware class out of all instances that the model classified as that class. It is calculated by dividing the number of true positive predictions (correctly classified instances of a particular class) by the sum of true positive and false positive predictions for that class.

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

3) *Recall:* Recall, also known as sensitivity, measures the proportion of correctly predicted instances of a specific malware class out of all actual instances of that class in the dataset. It is calculated by dividing the number of true positive predictions for that class by the sum of true positive and false negative predictions for that class.

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

4) *F1 Score*: The F1 Score is the harmonic mean of precision and recall, providing a balance between the two metrics, particularly useful when the dataset is imbalanced.

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

where, TP (True Positive): The number of correctly predicted instances of a specific malware family, indicating the model's ability to accurately classify that type of malware.

FP (False Positive): Instances where the model incorrectly classifies samples as a certain malware family when they actually belong to a different class or are benign.

TN (True Negative): The number of correctly identified samples that do not belong to the predicted class.

FN (False Negative): Instances where the model fails to identify samples that belong to a specific malware family, incorrectly classifying them as benign or as a different malware family.

E. Experiment Results

We divided the preprocessed dataset into training, validation, and testing sets, with 70% allocated for training, 15% for validation, and 15% for testing. This distribution ensures comprehensive training of the model, enabling effective hyperparameter tuning on the validation set, and robust evaluation of the model's ability to generalize to new, unseen data.

Additionally, we employed 10-fold cross-validation to further validate the experimental results and enhance the reliability of the model's performance.

Table 1 Performance Metrics of Each Model

Model	***SVM***	***RF***	***MLP***
Accuracy	**0.913**	**0.878**	**0.813**
Precision	**0.912**	**0.875**	**0.829**
Recall	**0.912**	**0.875**	**0.829**
F1-Score	**0.913**	**0.877**	**0.817**
Macro Avg	**0.91**	**0.88**	**0.817**
Weighted Avg	**0.91**	**0.88**	**0.817**

Table 1 summarizes the performance of three machine learning models: SVM, RF, and MLP. The metrics used to evaluate the models include accuracy, precision, recall, and F1-score. The SVM model outperforms both RF and MLP, achieving an accuracy of 91.3%, precision of 91.2%, recall of 91.2%, and an F1-score of 91.3%. The RF model follows with an accuracy of 87.8%, precision of 87.5%, recall of 87.5%, and an F1-score of 87.7%. The MLP model achieves an accuracy of 81.3%, precision of 82.9%, recall of 82.9%, and an F1-score of 81.7%.

In terms of the macro average, SVM scores 91%, RF scores 88%, and MLP scores 81.7%. The macro average calculates the metric independently for each class and then averages these values, treating each class equally regardless of its frequency in the dataset. This approach is particularly informative in cases where all classes are considered equally important, regardless of their prevalence.

Overall, the Support Vector Machine (SVM) is the best choice for now, demonstrating outstanding performance across all metrics.

IV. Way Forward

In this research, we demonstrated the effectiveness of using machine learning and Large Language Models (LLMs) to classify malware based on their dynamic behaviors. By leveraging LLMs for feature extraction, we improved the accuracy and robustness of traditional machine learning models like SVM, RF, and MLP. Our results showed that the Support Vector Machine (SVM) model outperformed the others, achieving the highest scores across all performance metrics. This study used a balanced and sufficiently large dataset of 20,000 samples across five malware categories, a significant improvement over our previous work, which used a smaller and less balanced dataset. Interestingly, the Multilayer Perceptron (MLP) model performed worse than in our earlier results. Although time and resource constraints limited our ability to test a larger neural network on the embedded malware behavior reports

generated by LLMs, we believe that a deeper neural network could significantly improve these outcomes.

Key challenges identified include the need for high-quality, relevant feature extraction by LLMs, the constant evolution of malware that necessitates continuous updates to detection models, and the importance of efficiently managing large datasets while maintaining high classification speeds. In this study, we explored different LLMs, such as BERT and GPT-2, for feature extraction, finding that BERT embeddings outperformed GPT-2 in terms of accuracy and robustness. This highlights the importance of selecting the right model for feature extraction, as the choice of LLM can significantly impact the effectiveness of malware detection systems.

Looking ahead, our future work will focus on utilizing the full CAPEv2 Dataset and developing an end-to-end deep learning model for malware classification that effectively addresses data imbalance. This will involve curating a more comprehensive set of malware samples to ensure broader coverage of malware types and behaviors, thereby improving the model's ability to generalize and accurately detect a wide range of threats. We also plan to deploy these models in real-world environments for real-time malware classification, including rigorous testing and fine-tuning in live settings to ensure they can handle the dynamic and fast-paced nature of actual cyber threats. Integrating these models with existing cybersecurity systems will enable immediate response and mitigation based on classification results, further enhancing the practical impact of our research.

To ensure the long-term effectiveness of our models, we will implement continuous learning and adaptation mechanisms. This includes developing strategies for the models to self-improve based on new data and feedback, keeping them updated with the latest malware trends. Performance optimization will be a key focus, with efforts to enhance computational efficiency and accuracy in managing large volumes of data swiftly and precisely. This will involve fine-tuning hyperparameters and employing advanced techniques to reduce false positives and negatives, ensuring our models remain resilient against evolving cyber threats. By detailing these steps, we aim to create a more robust and effective malware detection system capable of keeping pace with the fast-evolving threat landscape. The synergy between LLMs and dynamic behavior analysis offers a promising path forward in the fight against cyber threats, providing a comprehensive framework for strengthening cybersecurity defenses.

V. Acknowledgement

The authors would like to thank the Homeland Security Institute for funding and support in developing this technique paper.

VI. References

[1] Jiang, H.; Turki, T.; and Wang, J. T. L. 2018. Dlgraph: Malware detection using deep learning and graph embedding. In 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), 1029–1033.

[2] Nataraj, L.; Karthikeyan, S.; Jacob, G.; and Manjunath, B. S. 2011. Malware images: Visualization and automatic classification. In Proceedings of the 8th International Symposium on Visualization for Cyber Security, VizSec '11. New York, NY, USA: Association for Computing Machinery.

[3] Li, X., and Yu, H. 2021. Exploring the use of principal component analysis for reducing dimensionality in text embeddings. Journal of Machine Learning Research.

[4] Cao, J., and Gao, Y. 2020. Principal component analysis for efficient text embedding with gpt-2. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP).

[5] Singh, R., and Sharma, P. 2022. Dimensionality reduction of text embeddings using pca: A case study with gpt-2. IEEE Transactions on Neural Networks and Learning Systems.

[6] Chen, J.; Zhang, Y.; Li, S.; and Wu, X. 2020. Dynamic malware detection based on large language models. IEEE Access 8:190482–190492.

[7] Wang, T., and Li, M. 2021. Optimizing text embedding representations with principal component analysis and gpt-2. ACM Transactions on Information Systems.

[8] Bosansky, B.; Kouba, D.; Manhal, O.; Sick, T.; Lisy, V.; Kroustek, J.; and Somol, P. 2022. Avast-ctu public cape dataset.

[9] Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In Burstein, J.; Doran, C.; and Solorio, T., eds., Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

[10] Alqahtani, S., and Jones, J. A. 2021. Dynamic malware analysis using machine learning techniques. In Proceedings of the 2021 IEEE International Conference on Big Data (Big Data), 5199–5202. IEEE.

[11] Dahl, G. E.; Stokes, J. W.; Deng, L.; and Yu, D. 2013. Large-scale malware classification using random projections and neural networks. In 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 3422–3426. IEEE.

[12] Hardy, W.; Chen, L.; Hou, S.; Ye, Y.; and Li, X. 2016. Dl4md: A deep learning framework for intelligent malware detection. In Proceedings of the International Conference on Data Mining, 61–78. Springer.

[13] Khan, M. M.; Abbas, G.; and Mehmood, Z. 2022. Random forest-based detection of evolving malware variants. IEEE Access 10:33498–33510.

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

[15] Sharma, R., and Dash, D. 2021. Malware classification using deep learning techniques. In Proceedings of the 2021 International Conference on Advances in Computing, Communication, and Control (ICAC3), 284–290.

Authors biography

Dr. Haodi Jiang currently is an Assistant Professor in the Department of Computer Science at SHSU. He earned his Ph.D. in Computer Science from the New Jersey Institute of Technology (NJIT). His research interests include machine learning, artificial intelligence, computer vision, with applications in solar physics, space weather, cybersecurity, and bioinformatics. His work has been published in high impact journals (e.g., ApJS, ApJ, Sol. Phys), biomedical imaging journals (e.g., CMIG), data mining journals (e.g., IDA), and machine learning and data mining conferences (e.g., ICTAI, ICMLA). Additionally, Dr. Jiang has served as a NASA panelist and a reviewer for major journals and conferences (e.g., Nature Astronomy, Astronomy & Astrophysics, IEEE Transactions on Cybernetics, TKDD, ICDM, etc.).

Tosin B. Akinsowon is currently a Doctoral Research Assistant in the Department of Computer Science at Sam Houston State University, pursuing a Ph.D. in Digital and Cyber Forensic Science with a GPA of 4.0. He holds a Master's in Policing from the Criminal Investigation Police University of China and both a B.Sc. and M.Sc. in Computer Science from the University of Lagos, Nigeria. With nearly a decade of experience as a Cybercrime Intelligence Officer at INTERPOL NCB Abuja, he specializes in intelligence-driven policing, white-collar crime, anti-fraud, and digital forensics. Tosin's research interests include machine learning, malware analysis, and digital forensics, and he actively contributes to the cybersecurity field through his work. He is a member of professional organizations such as IEEE and the Interpol Cyber Security Expert Group and mentors for the Women in Cyber Program.

INSTITUTE FOR HOMELAND SECURITY

Sam Houston State University

The Institute for Homeland Security at Sam Houston State University is focused on building strategic partnerships between public and private organizations through education and applied research ventures in the critical infrastructure sectors of Transportation, Energy, Chemical, Healthcare, and Public Health.

The Institute is a center for strategic thought with the goal of contributing to the security, resilience, and business continuity of these sectors from a Texas Homeland Security perspective. This is accomplished by facilitating collaboration activities, offering education programs, and conducting research to enhance the skills of practitioners specific to natural and human caused Homeland Security events.

Institute for Homeland Security
Sam Houston State University

Akinsowon, T., & Jiang, H. (2024). Leveraging large language models for behavior-based malware detection using deep learning (Report No. IHS/CR-2024-1035). Sam Houston State University, Institute for Homeland Security.
https://doi.org/10.17605/OSF.IO/YG62X