International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 9, Issue 3, pp 221-227, March-2025

https://doi.org/10.47001/IRJIET/2025.903029

Enhanced Fileless Malware Detection Using a Deep Learning Approach

[1]*Seema B Joshi, [2]Rohita Regunathan Warrier

[1]Gujarat Technological University, Ahmedabad, 382424, Gujarat, India

[2]ME - Cybersecurity (Batch-2022-24) PG Scholar, Gujarat Technological University, Ahmedabad, 382424, Gujarat, India

***Abstract* - This study addresses the escalating threat of fileless malware, which bypasses traditional cybersecurity measures by operating exclusively in volatile memory, posing a formidable challenge to detection. Through the integration of memory forensics and deep learning, we introduce an innovative method to improve fileless malware detection. Leveraging memory dump analysis, we extract unique characteristics and patterns associated with fileless malware, employing deep learning algorithms tailored for this purpose. The research aims to create a strong detection framework for accurately identifying fileless malware, which is essential for enhancing cybersecurity resilience. Motivated by the urgency to combat evolving cyber threats, our study focuses on developing and validating a dataset derived from memory forensics and applying deep learning algorithms for malware detection. We employ specialized tools such as Magnet RAM Capture and the Volatility Framework to acquire memory dumps and extract relevant features. Fileless malware samples are collected and executed within a controlled environment, with their memory dump features used to build a comprehensive dataset. Deep learning classifiers, including recurrent neural networks (RNNs) and deep neural networks (DNNs), are chosen for binary classification of fileless malware. The DNN model demonstrates exceptional performance, achieving an accuracy of 97.58% with a true positive rate (TPR) of 97.05% and a minimal false positive rate (FPR). This underscores the efficacy of deep learning in accurately detecting fileless malware, particularly in identifying malicious activities rather than relying on file signatures or registry entries. In the evolving threat landscape, deep learning models provide scalability and efficiency in managing large and diverse datasets, making them essential for combating fileless malware.**

Keywords: Fileless malware, deep learning, memory dump analysis, feature engineering, malware detection.

I. INTRODUCTION

In recent years, the proliferation of fileless malware has posed a significant challenge to traditional cybersecurity measures, evading detection by relying on volatile memory rather than static files on disk. This emerging threat underscores the critical need for innovative approaches to malware detection and mitigation. As malware attacks become more sophisticated, leveraging techniques like phishing and script-based infiltration, existing detection methods based on file signatures or heuristic analysis have shown limitations in effectively identifying fileless malware. The ability of fileless malware to operate stealthily within a system's memory without leaving traces on disk presents unique challenges for cybersecurity professionals.

To address these challenges, this study explores the convergence of memory forensics and deep learning for enhanced fileless malware detection.

Leveraging memory dump analysis, which captures the volatile state of a system's memory, we aim to extract distinct characteristics and patterns associated with fileless malware. By harnessing deep learning algorithms specifically tailored to process and analyze these extracted features, we seek to develop a robust detection framework capable of identifying fileless malware with high accuracy and efficiency. This research contributes to bridging the gap in cybersecurity by proposing a novel approach that combines advanced memory analysis with state-of-the-art deep learning techniques.

The study was motivated by the critical necessity of enhancing cybersecurity resilience against fileless malware threats. Our approach is designed to advance understanding and detection capabilities in this domain, ultimately aiming to develop effective strategies for combating this evolving cybersecurity menace. The objectives of this research include developing and validating a dataset derived from memory forensic techniques, applying deep learning algorithms to detect fileless malware, and assessing the performance of these methods against real-world scenarios. Through this work, we aim to provide a comprehensive and practical framework that addresses critical gaps in current malware detection methodologies and contributes to the ongoing evolution of cybersecurity defense.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 9, Issue 3, pp 221-227, March-2025

https://doi.org/10.47001/IRJIET/2025.903029

II. RELATED WORK

2.1 Malware Detection

Currently, malware is evolving with sophisticated techniques to launch cyberattacks, transitioning from traditional file-based methods to fileless attacks to evade existing detection solutions [13]. While existing solutions [14, 15] effectively identify file-based malware attacks on Windows [16], Android [17, 18], and IoT devices [19], they struggle to detect fileless malware. This section provides a literature review and comparative analysis focusing on machine learning approaches tailored to detecting fileless malware.

Lee et al. [8] conducted a comprehensive analysis of ten previously known fileless malware strains sourced from public websites like hybrid-analysis and GitHub. They came up with a way to group things into groups based on attack methods and traits, using Cuckoo Sandbox to look at malware and connecting it to the MITRE ATT&CK kill chain [20]. Their work categorized fileless malware into three groups (attack type, evasion, and data collection) based on results from Cuckoo Sandbox, enhancing response times to fileless attacks.

Osama Khalid et al. described a method for detecting and analyzing fileless malware using a combination of memory forensics and machine learning techniques. The author explained that memory forensics involves analyzing the contents of a computer's memory (a memory dump) to uncover evidence of malicious activity. By creating a snapshot of the infected machine's memory, one can use specialized tools like Volatility to extract features related to fileless malware. After extracting relevant features from the memory dump using tools like Volatility, the author proposed using machine learning to build a model capable of detecting fileless malware. Their model is trained on the extracted features to recognize patterns associated with fileless malware behavior.

Afreen et al. [7] studied different types of fileless malware and how they infect systems using various techniques like.NET frameworks, Windows Management Instrumentation, and PowerShell. They discussed how fileless malware attacks occur without leaving traditional traces, like infected files, and can gain persistence on systems. They emphasized the importance of enhancing analysis methods to improve fileless malware detection. Additionally, they highlighted that fileless malware can infect systems not only through infected files but also via web browsers.

2.2 Memory Forensics

In computer forensics, two primary approaches are used for analyzing potentially malicious software: dynamic analysis and static analysis. Static analysis is conducted without executing or running the program being analyzed. Instead, it involves examining the code, binaries, or artifacts of the software to extract information about its behavior, structure, or potential threats. Extracting and examining the components, instructions, and control flow of the program without executing it. This could involve disassembling or decompiling binaries to understand the underlying code.

III. METHODOLOGY

As shown in figure 1, The methodology for malware detection using memory forensics combined with deep learning models involves a systematic approach to extracting, analyzing, and classifying features from memory dumps to identify fileless malware. This step focuses on monitoring and analyzing system behaviors to detect anomalies indicative of malicious activity. Features are extracted from memory dumps to capture critical artifacts like running processes, loaded DLLs, network connections, and registry entries using tools like Volatility and AnyRun.

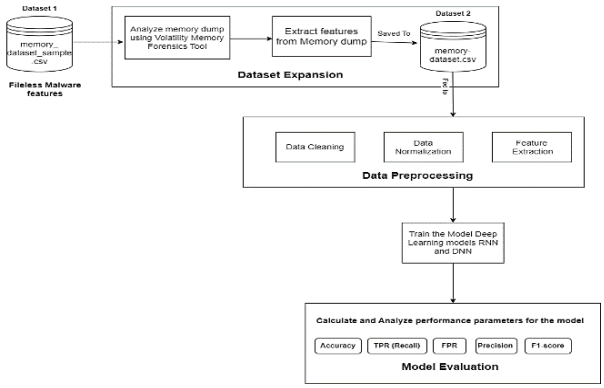

Figure 1: Flowchart of the Proposed Methodology

Memory dumps are acquired using specialized tools like magnet RAM capture, which captures the volatile memory state of a target system. The acquisition settings are configured to specify the memory range and output location for the memory dump. The Volatility Framework is employed to extract relevant characteristics from memory dumps, utilizing over 70 plugins to analyze various memory artifacts. Plugins retrieve information such as process lists, loaded DLLs, running services, and network connections.

Fileless malware samples are collected from specific sources and executed within a virtual machine environment to trigger their behaviors. Memory dumps from executed malware samples are analyzed to extract features for building the dataset. Additional fileless malware samples are added to balance the dataset. Deep learning classifiers like recurrent neural networks (RNNs), deep neural networks (DNNs), and their combinations are chosen for binary classification of

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 9, Issue 3, pp 221-227, March-2025

https://doi.org/10.47001/IRJIET/2025.903029

fileless malware. RNNs, particularly LSTM or GRU architectures, are suitable for processing sequential data and capturing temporal patterns in memory dump features. DNNs, including feedforward networks and CNNs, excel at extracting complex, nonlinear relationships between memory artifacts for accurate classification.

IV. MALWARE DETECTION APPROACH

Malware detection using memory forensics combined with deep learning models represents an advanced approach to identifying and analysing fileless malware. This method involves analysing the contents of a computer's memory (memory dump) to extract features indicative of malicious activity and then leveraging deep learning algorithms for detection. Here's an overview of this approach, highlighting the limitations of traditional machine learning models in contrast to deep learning.

Traditional machine learning often requires extensive manual feature engineering, where domain expertise is needed to handcraft relevant features from the data. This process can be time-consuming and may overlook complex patterns that deep learning can capture automatically. Traditional machine learning models, which typically assume linear relationships between features and outputs, may not be able to adequately capture the complex and nonlinear behaviours that fileless malware frequently exhibits. So, we will use deep learning models with memory analysis in order to detect fileless malware.

4.1 Behaviour Analysis and Features Extraction

Behaviour analysis and feature extraction are fundamental techniques in cybersecurity used to identify and characterize malicious activities or anomalies within computer systems. These techniques are crucial for detecting sophisticated threats, such as malware, intrusions, or unauthorized activities.

Behaviour analysis involves monitoring and analysing the actions and interactions of entities (e.g., processes, users, and network traffic) within a system to identify patterns indicative of malicious behaviour. The goal is to understand normal behaviour and detect deviations or suspicious activities that may indicate a security threat.

4.2 Acquisition of memory dumps from the RAM capture tools

Acquiring a memory dump from RAM is a crucial step in memory forensics for extracting features and conducting malware detection. This process involves capturing the contents of a computer's volatile memory (RAM) to create a snapshot of the system's state at a specific point in time. Memory dump acquisition tools are used to facilitate this process efficiently and accurately. Memory dump acquisition enables the extraction of features necessary for malware detection and analysis. By analysing memory artifacts, security analysts can identify indicators of compromise (IOCs), unusual process behaviours, code injections, and other malicious activities associated with malware.

After acquiring a memory dump, security analysts identify relevant artifacts and data structures for feature extraction. Key artifacts include process memory regions, loaded modules, registry hives, network connections, and system call traces. Memory dump features serve as input for machine learning and deep learning models used in automated malware detection. These models learn to differentiate between benign and malicious behaviours based on extracted features, enhancing the accuracy of malware detection.

Magnet RAM Capture is a specialized tool used for acquiring memory dumps from volatile memory (RAM) in live computer systems. This tool is designed for digital forensics and incident response professionals to capture and preserve the state of a system's memory for subsequent analysis and investigation. In the proposed model, we have used the Magnet RAM Capture tool to acquire memory dumps. Following is the process to collect the memory dump using the Magnet RAM Capture tool:

- Before performing memory acquisition, ensure that Magnet RAM Capture is installed on the forensic workstation or investigation system. Prepare the target system for memory capture by running the tool with appropriate permissions and access rights.
- Launch Magnet RAM Capture and configure acquisition settings, such as specifying the target system, desired memory range, and output location for the memory dump.
- Start the memory capture process using Magnet RAM Capture. The tool interacts with the target system's memory management subsystem to capture the contents of physical RAM and virtual memory.

4.3 Feature Extraction from the Memory Dumps

After acquiring the memory dump from a virtual machine, the next step involves extracting features from the memory dump to analyse malware or non-malware samples. These extracted features are saved into a CSV file for use in training and testing machine learning models. The Volatility Framework tool is utilized for feature extraction, offering a comprehensive suite of over 70 plugins designed to analyse various characteristics of main memory. Volatility supports

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 9, Issue 3, pp 221-227, March-2025
https://doi.org/10.47001/IRJIET/2025.903029

both 32-bit and 64-bit operating systems, including Windows, Linux, and macOS variants.

To begin analysing the memory dump, it's essential to initialize the appropriate profile in Volatility that corresponds to the operating system of the memory dump. This profile helps Volatility interpret the memory dump correctly. Once the profile is set, different Volatility plugins can be executed to extract valuable information from the memory dump. These plugins can retrieve details such as the list of running processes, loaded DLLs by processes, running services, network connections, and registry hives.

While volatility is powerful for many analyses, it may not provide certain details, such as registry events and specific network information like DNS requests. To supplement this information, the malware sample can be executed in an online sandbox environment like AnyRun.

AnyRun captures additional network- and registry-related features generated by the malware sample during execution.

In summary, the feature extraction process involves using volatility to analyse the memory dump and extract relevant characteristics and behaviours of the system. Additional network and registry-related features are obtained by executing the malware sample in a sandbox environment. These extracted features are crucial for building and evaluating machine learning models for malware detection and analysis.

4.4 Dataset and Fileless Malware Sample Details

Fileless malware samples are collected from specific websites. These samples are sourced from known sources of malware to study their behaviour and characteristics. Each collected fileless malware sample is executed one by one within a virtual machine environment. The goal is to trigger the execution of the fileless malware and capture the resulting memory dump from the virtual machine.

After acquiring the memory dump from the virtual machine, features are extracted using the Volatility Memory Forensics tool. Volatility is used to analyse the memory dump and extract relevant characteristics, such as running processes, loaded DLLs, network connections, registry hives, and other system artifacts. The initial dataset obtained from [32] is described as unbalanced, with slightly more non-malware samples compared to malware samples.

To balance the dataset, 26 new fileless malware samples are added. Among the newly added fileless malware samples, only five are successfully executed on the virtual machine, while the remaining 21 fail to execute due to inactive command and control servers.

The final dataset used for the study consists of a balanced set of fileless malware and non-malware samples. A set of extracted features obtained from each sample's corresponding memory dump using volatility serves as its representation in the dataset.

These features serve as input data for training and evaluating machine learning models for malware detection. Table 1 shows features of fileless malware that differentiate them from file-based malware. Table 2 shows the advanced Features extracted to detect fileless malware.

Table 1: Features that differentiate file-less malware from file-based malware

Feature	Command	Description
API	Impscan	Malware use API calls to communicate with the operating system.
DLL	Dlllist	Malware inject DLLs aiming to insert malicious code into a legitimate process.
Process Handle	Handles	Give information about malware behaviors, such as reading from a file, writing to a file, and accessing a registry key.
Privilege	Privs	Permission is giving to a process to do specific tasks, like changing the time, loading kernel drive, and shutting down the computer.
Network	Netscan	It is used to communicate with the attacker to receive commands or send user information.
Code Injection	Malfind	The malware attempts to inject its malicious code into another legitimate process to force the latest to runcode on its behalf.

Table 2: Advanced Features extracted to detect fileless malware

Feature	Command	Description
Loaded modules	ldrmodules	Lists loaded modules, including injected DLLs.
Running services.	getservicesids	Lists the security identifiers (SIDs) for running services.
API hooks	apihooks	Identify API hooks and detours in memory
callbacks	callbacks	Used to identify and analyze registered callbacks in the system, which might be leveraged by fileless malware.

4.5 Selection of Classifiers

Deep learning algorithms like recurrent neural networks (RNNs), deep neural networks (DNNs), and combinations of both are very good at binary class classification for finding malware without files. These algorithms excel at learning intricate patterns and representations from raw data, making them suitable for tasks requiring complex feature extraction and classification. RNNs are particularly effective for processing sequential data, making them well-suited for analyzing temporal patterns in memory dump features. In the context of malware detection, RNNs can capture dependencies and relationships between different memory artifacts over

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 9, Issue 3, pp 221-227, March-2025

https://doi.org/10.47001/IRJIET/2025.903029

time, aiding in the identification of anomalous behaviors associated with fileless malware. RNN architectures, such as Long Short-Term Memory (LSTM) or Gated Recurrent Units (GRUs), are often used to model memory dump sequences and pull out features that make them different for classification.

DNNs are powerful models that can automatically learn hierarchical representations from complex input data. In the context of feature extraction from memory dumps, DNNs can process multi-dimensional feature vectors derived from various memory artifacts. DNN architectures, like deep feedforward networks or convolutional neural networks (CNNs), can effectively capture non-linear relationships between memory dump features. This makes it easier to tell the difference between malware and non-malware samples.

V. EXPERIMENTAL EVALUATION

5.1 Feature Scaling

Feature scaling is a critical preprocessing step in deep learning models such as recurrent neural networks (RNNs) and deep neural networks (DNNs) for fileless malware detection. Feature scaling involves normalizing or standardizing input features to ensure that they have a consistent scale and distribution, which can significantly improve the performance and convergence of the models during training. In the context of memory dump analysis for fileless malware detection, RNNs process sequences of memory artifacts (e.g., process information, DLLs, and network connections) over time. Feature scaling is essential for RNNs because it helps mitigate the issue of vanishing or exploding gradients, which can occur when input features have varying scales. Common approaches to feature scaling in RNNs include min-max scaling (normalization) and standardization (subtracting the mean and dividing by the standard deviation). By scaling input features to a consistent range (e.g., [0, 1] for normalization or z-score for standardization), RNNs can effectively learn sequential patterns and dependencies in memory dump sequences without being biased by the magnitude of individual features.

5.2 Parameter Optimization

Parameter optimization in the context of deep learning models involves the process of tuning hyperparameters and model configurations to enhance performance, improve convergence, and achieve optimal results for a specific task, such as fileless malware detection. Deep learning models consist of various parameters, including network architecture, activation functions, learning rates, batch sizes, and regularization techniques, among others. The goal of parameter optimization is to find the best combination of these parameters that maximizes the model's predictive accuracy and generalization ability on unseen data. It includes hyperparameter tuning, model configuration, and cross-validation.

VI. RESULTS AND DISCUSSIONS

The outcomes are presented based on the true positive rate (TPR), false positive rate (FPR), and accuracy score. TPR signifies the proportion of correctly identified malware samples out of all actual malware samples. FPR represents the proportion of non-malware samples incorrectly classified as malware. Accuracy denotes the ratio of correct classifications to the total number of classification attempts.

We compare the performance of selected classifiers using default parameters on original, unscaled features. Here, we display the results obtained using the same original features, but with classifiers trained on optimal parameters. Notably, some classifiers, like Support Vector Machines (SVM), exhibit improved performance on scaled data due to their sensitivity to variance. However, classifiers such as decision trees (DT) and RF perform consistently well regardless of scaling. Throughout our experiments, the RNN+DNN model demonstrated robust performance across different settings, maintaining a high accuracy of 97.58%, a TPR of 96.07%, and a 0.008% FPR. It's important to highlight that although RNN achieved a comparable TPR of 97.5% when optimized, it incurred a higher FPR of 2.01% compared to RF's 0.008% FPR.

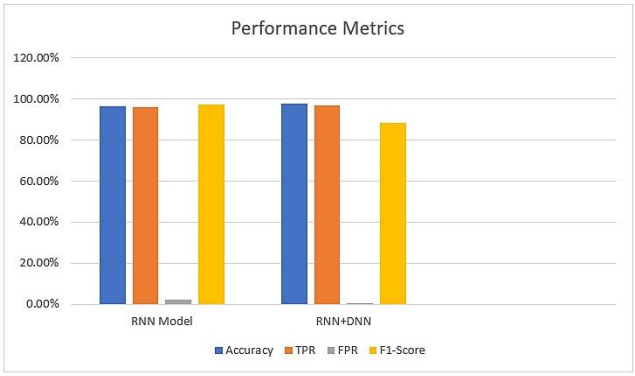

Figure 2: Performance metrics

The RNN+DNN (proposed) model shows exceptional performance in malware detection, achieving high accuracy, precision, recall, and F1-score with the lowest FPR as illustrated in figure 2. The RNN+DNN model slightly improves accuracy to 97.58% with balanced precision and recall. It has the lowest FPR, which makes it more efficient at detecting malware. The RNN model achieves a decent accuracy of 96.34% with good precision and recall for both benign and malicious classes. But, again, its FPR is slightly higher than the proposed model. This higher FPR leads to

ISSN (online): 2581-3048
Volume 9, Issue 3, pp 221-227, March-2025
https://doi.org/10.47001/IRJIET/2025.903029

lower accuracy. A high FPR reduces specificity because the model fails to correctly identify a substantial number of negative instances. The RNN model achieved an accuracy of 0.967, or 96.34%, with a TPR of 96.07% and a FPR of 2.01%.

VII. CONCLUSION AND FUTURE WORK

Fileless malware represents a formidable cybersecurity challenge by eschewing traditional installation on a system's hard drive and instead operating directly in the main memory (RAM). Its stealthy nature, leveraging legitimate programs and tools, makes detection challenging as it leaves behind no typical traces like files or registry entries. The focus of this malware detection approach is on identifying malicious activities rather than depending on the presence of malware files. Deep learning techniques, particularly the Deep Neural Network, have proven effective in this context. In these experiments, the DNN model exhibited outstanding performance with an accuracy of 97.58%. Notably, it achieved a true positive rate (TPR) of 97.05% while maintaining a very low false positive rate (FPR) on an unseen test set. These metrics underscore the model's precision in correctly identifying instances of fileless malware and distinguishing them from non-malicious (benign) samples.

Deep learning models exhibit high scalability and efficiency in managing large datasets. With the increasing volume and diversity of malware samples, deep learning models can utilize parallel processing on GPUs or TPUs to accelerate model training and inference. Thus, deep learning models are highly efficient for detecting the memory features of fileless malware.

Further refine feature extraction techniques from memory dumps to enhance the discriminative power of deep learning models. Investigate advanced feature representation methods, including embeddings and attention mechanisms, to highlight critical aspects of memory dump sequences. Explore techniques for generating synthetic data to augment existing datasets, addressing challenges associated with limited and imbalanced datasets in fileless malware detection. Apply transfer learning techniques by leveraging pre-trained models on related tasks (e.g., general malware detection or memory forensics) to improve the performance of fileless malware detection models.

CONFLICTS OF INTEREST

The authors have no conflicts of interest to declare.

REFERENCES

[1] Khalid O, Ullah S, Ahmad T, Saeed S, Alabbad DA, Aslam M, et al. An insight into the machine-learning-based fileless malware detection. *Sensors (Basel) [Internet]*. 2023;23(2). Available from: http://dx.doi.org/10.3390/s23020612.

[2] Usman N, Usman S, Khan F, Jan MA, Sajid A, Alazab M, et al. Intelligent, dynamic malware detection using machine learning in IP reputation for forensic data analytics. *Future Generation Computer Systems*. 2021; 118:124–41.

[3] Shah SSH, Ahmad AR, Jamil N, Khan A ur R. Memory forensics-based malware detection using computer vision and Machine Learning. *Electronics (Basel) [Internet]*. 2022;11(16):2579. Available from: http://dx.doi.org/10.3390/electronics11162579.

[4] Bozkir AS, Tahillioglu E, Aydos M, Kara I. Catch them alive. A malware detection approach through memory forensics, manifold learning, and computer vision. *Computers & Security*. 2021;103.

[5] Zhang S, Hu C, Wang L, Mihaljevic MJ, Xu S, Lan T. A Malware Detection Approach Based on Deep Learning and Memory Forensics. *Symmetry*. 2023;15(3).

[6] Ayad A, Farag HEZ, Youssef A, El-Saadany EF. Detection of false data injection attacks in smart grids using Recurrent Neural Networks. *In: 2018 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT). IEEE*; 2018.

[7] Liu J-D, Ou Y-Y. An improved XSS vulnerability detection method based on attack vector. *DEStech Trans Comput Sci Eng [Internet]. 2018;(icmsa).* Available from: http://dx.doi.org/10.12783/dtcse/icmsa2018/23251.

[8] Ahmadi A, Nabipour M, Taheri S, Mohammadi-Ivatloo B, Vahidinasab V. A new false data injection attack detection model for cyberattack resilient energy forecasting. *IEEE Trans Industr Inform [Internet]*. 2023;19(1):371–81. Available from: http://dx.doi.org/10.1109/tii.2022.3151748.

[9] Pei C, Xiao Y, Liang W, Han X. PMU placement protection against coordinated false data injection attacks in smart grid. *IEEE Trans Ind Appl [Internet]*. 2020;56(2):1–1. Available from: https://yangxiao.cs.ua.edu/PMU_Placement_Protection_Against_Coordinated_False_Data_Injection_Attacks_in_Smart_Grid.pdf

[10] Huang K, Xiang Z, Deng W, Yang C, Wang Z. False data injection attacks detection in smart grid: A structural sparse matrix separation method. *IEEE Trans Netw Sci Eng [Internet]*. 2021;8(3):2545–58. Available from: http://dx.doi.org/10.1109/tnse.2021.3098738.

[11] Roy P, Kumar R, Rani P. SQL injection attack detection by machine learning classifier. *In: 2022*

ISSN (online): 2581-3048
Volume 9, Issue 3, pp 221-227, March-2025
https://doi.org/10.47001/IRJIET/2025.903029

International Conference on Applied Artificial Intelligence and Computing (ICAAIC). IEEE; 2022.

[12] Singh SK, Khanna K, Bose R, Panigrahi BK, Joshi A. Joint-transformation-based detection of false data injection attacks in smart grid. *IEEE Trans Industr Inform [Internet]*. 2018;14(1):89–97. Available from: http://dx.doi.org/10.1109/tii.2017.2720726.

[13] Tripathy D, Gohil R, Halabi T. Detecting SQL injection attacks in cloud SaaS using machine learning. *In: 2020 IEEE 6th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS). IEEE;* 2020.

[14] Ojagbule O, Wimmer H, Haddad RJ. Vulnerability analysis of content management systems to SQL injection using SQLMAP. *In: SoutheastCon 2018. IEEE;* 2018.

[15] Scott D, Sharp R. Abstracting application-level web security. *In: Proceedings of the 11th international conference on World Wide Web. New York, NY, USA: ACM;* 2002.

AUTHORS BIOGRAPHY

Dr. Seema B Joshi is an Assistant Professor (Cyber Security) at Gujarat Technological University, Ahmedabad, Gujarat, India with 15 years of experience successfully contributing to cyber security curriculum development and delivery. Driven to contribute to programme outcomes by facilitating engagement and supporting learning objectives. Enthusiastic professional with a background in academic advice. Passionate academician and researcher about fostering academic development and success for every student. Email: ap_seema@gtu.edu.in

Ms. Rohita Warrier was the PG scholar of ME CE Cyber Security (Batch 2022-24) at Gujarat Technological University - School of Engineering and Technology, Ahmedabad, Gujarat, India. The malware analysis and research were her dissertation topic during her master's study.

Citation of this Article:

Seema B Joshi, & Rohita Regunathan Warrier. (2025). Enhanced Fileless Malware Detection Using a Deep Learning Approach. *International Research Journal of Innovations in Engineering and Technology - IRJIET*, 9(3), 221-227. Article DOI https://doi.org/10.47001/IRJIET/2025.903029

****

