IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, September 2026
Impact Factor: 8.2

# Deep Learning-Based Sequence Analysis for Early Detection of Ransomware Behavior in Cloud Environments

**Sanjay Kumar Maurya[1] and Prof. (Dr.) Upma Sharma[2]**
[1] M.Tech Scholar, [2] Professor, Department of Computer Engineering & Technology,
HRIT University, Ghaziabad

**Abstract:** *The rapid growth of cloud computing and Internet of Things (IoT) environments has increased the complexity and frequency of malware and spyware attacks, creating significant challenges for reliable and adaptive threat detection. Conventional malware detection approaches often suffer from limited adaptability, privacy concerns, inadequate feature representation, and difficulties in detecting evolving attack patterns. To address these challenges, this study proposes a D2EBTN (Deep Dual-Training Ensemble-Based Temporal Network) framework for efficient malware identification in cloud environments. The proposed framework utilizes data from three publicly available datasets, namely CIC-MalMem-2022, CIC IoT Dataset 2023, and CIC-IDS2017. During preprocessing, redundant attributes are removed and missing values are appropriately imputed to improve data quality and learning efficiency. The proposed architecture integrates Incremental Learning (IL) and Federated Learning (FL) to provide adaptive and privacy-preserving malware detection. IL enables the model to learn from newly available tasks while retaining knowledge acquired from previous tasks, thereby reducing catastrophic forgetting. FL facilitates collaborative learning through multiple local models and a global model without directly sharing sensitive data. The core learning architecture incorporates three Bi-LSTM models, a gated multidirectional attention mechanism, and a Deep Belief Network (DBN) for temporal feature learning, significant feature extraction, attention-based representation, and final malware classification. The proposed framework is comparatively evaluated against GCRD, CGOIDL-MD, FL-DNN, AutoML-CNN, Me-Mal-Det, and XGBoost using common evaluation metrics. Experimental results on the CIC-MalMem-2022 dataset demonstrate that D2EBTN achieves superior performance at a 90% training level in terms of accuracy, F1-score, precision, sensitivity, and selectivity. The model provides precision improvements of 0.34%, 1.89%, and 1.63% over GCRD, CGOIDL-MD, and FL-DNN, respectively, while further improvements are observed over AutoML-CNN, Me-Mal-Det, and XGBoost. These findings demonstrate that integrating dual learning with attention-based temporal feature extraction can provide an effective, adaptive, and privacy-aware solution for malware detection in dynamic cloud environments.*

**Keywords**: Cloud Malware Detection; Incremental Learning; Federated Learning; Bi-LSTM; Gated Multidirectional Attention

## I. INTRODUCTION

As computing technologies become more cloud-based, computer security is expected to become increasingly important, as it offers many advantages over traditional security strategies. Many computing services are available via the cloud over the Internet for users and businesses [1]. Instead of users storing data locally on their computers, data can be stored remotely in data centers. Figure 1 shows an example of a typical cloud computing infrastructure and environment. In a cloud computing environment, you can store data, run servers, run virtual machines, build databases, connect to networks, and download software.


DOI: 10.48175/568

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, September 2026
Impact Factor: 8.2

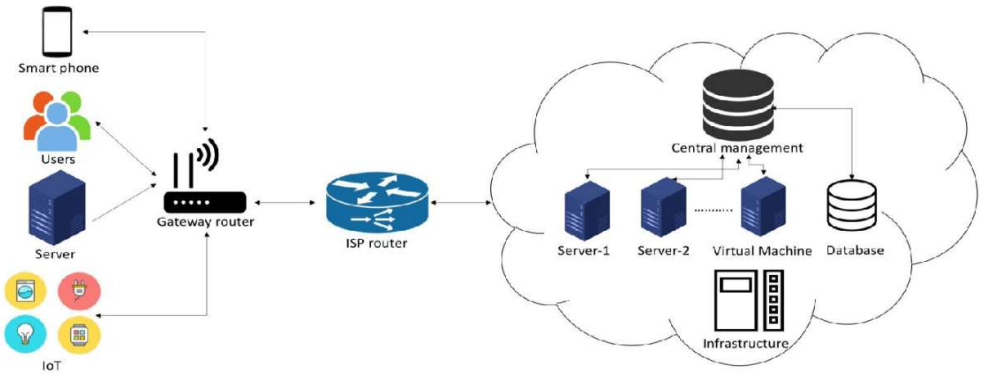


**Figure 1: Cloud Computing (CC) Infrastructure**

CC service models are represented in figure 2 as a service model, software is delivered through SaaS, infrastructure is delivered through IaaS, and platforms are delivered through PaaS. There are various benefits and business proposals associated with each of these cloud models. SaaS has many advantages, but it also has some drawbacks. In IaaS, virtual machines (VMs) are used to provide infrastructure; however, they are wasting resources. In the cloud, virtual machines are not adequately protected, and the infrastructure isn't well suited to provide security. Data can be deleted within a specified time frame; Clients and cloud providers can both determine this. Using IaaS may pose a compatibility issue because client-only operating systems run legacy software [2]. Shared physical resources can only be achieved by dividing virtual machines between secure hypervisors.

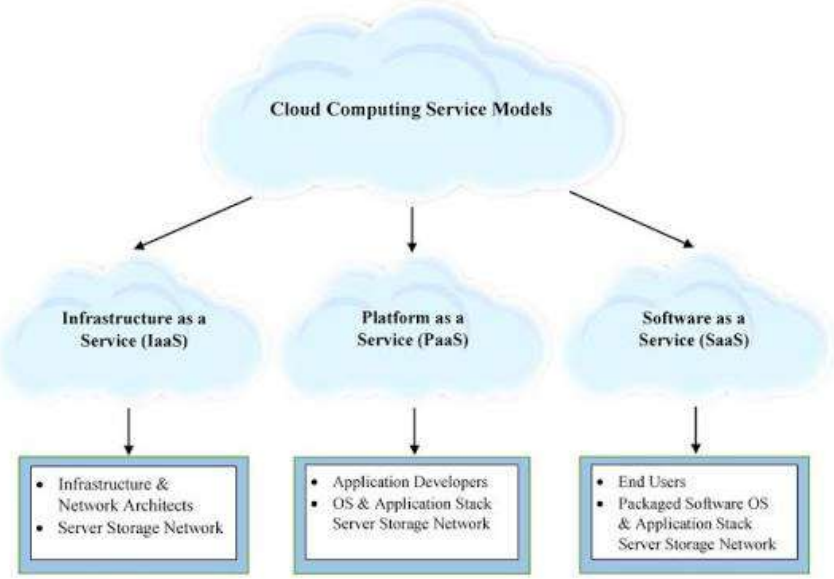


**Figure 2: Cloud Computing (CC) Service Models**

Developers can develop and deploy software using PaaS by using a web-based service [2]. A number of issues are involved, the interoperability of the system, host vulnerabilities, private authentication, fault tolerance, and continuity of the service are all taken into account. Direct deployments are not required with SaaS. After all, since it is geographically dispersed and delivers near-real-time. There are several important SaaS security considerations, including data privacy, availability, and network security [3]. Hackers execute malware whenever they succeed in deceiving a cloud system. By manipulating or stealing data from this network, a malicious attacker can spy on it [4].

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, September 2026
Impact Factor: 8.2

## II. RELATED WORK

In the January 2002 edition of Virus Bulletin journal [3], the problem of naming was explored and it is concluded that while some antivirus companies do not use CARO names extensively, there is a lack of vendor collaboration leading to the use of various malware names as aliases. The full-text description in addition to the naming convention levels make up the third level of information that is used. In the absence of a generally accepted naming convention, antivirus providers employ their own. The names are somewhat different even if they have similar names. The identically same malware instance has at least four different names in a 2002 Symantec publication, which increases the redundancy for instances of the same infection. It's not immediately relevant to the current task, but it has to be considered in order to ensure that the categorization findings are not impacted by the name issue. The authors of [4] look at how machine learning methods are used to identify malware on Android. To identify malware on Android devices, we created a mathematical method that relies on machine learning techniques and static feature analysis of Android apps.

Three sorts of settings may be utilized to evaluate malware detection models: the hybrid analysis combines static and dynamic analytic approaches. Four factors were tested using static analysis [9]. Several analytical approaches can be applied, including code type verification, program slicing, taint assessment, symbolic code execution, and abstract interpretation. The sensitivity analysis would consider the following elements: objects, contexts, fields, pathways, and flows. There are three layers of analysis: emulator, application, and kernel. The dynamic analysis technique may be used to analyze taints and anomalies.

There are two types of feature extraction strategies available in static analysis. Examination of Codes and Manifestations [10] Manifest analysis can reveal elements such as package name, activities, intentions, permissions, and service providers. Examining the code's information flow, opcode, native code, API calls, and taint tracking shows a number of useful characteristics. It was determined that there were five effective strategies for obtaining features from dynamic analyses. Network traffic, Java classes, and intents were all monitored. (1) Network traffic analysis involved identifying IP addresses, URLs, certificates, network protocols, and unencrypted data. (2) Network taffic, JAVA classes, and intent were instrumented by code instrument analysis. (3) Analysis of system calls (4) Button, icon, and action/event user interaction analysis, (5) Process reports, network usage, and resource utilization of CPUs, memory, and batteries by System Resource analysis. Next, a hybrid model that incorporates both methods is created to enhance malware identification and detection. The separate nature of static and dynamic analysis means that hybrid analysis methodologies are more resource-intensive and take longer to complete.

The hybrid analysis used permission intent and user behavior to identify malware most frequently. As an extra precaution, hybrid characteristics are employed to improve the detecting system's accuracy. The information shown prior to the download and installation of a program is referred to as an application metadata feature. N-grams, Opcodes, URLs, API requests, IP, Network Protocol, and so on are all included.

## III. PROPOSED METHODOLOGY

The study seeks to provide a D2EBTN framework for efficient spyware identification in cloud environments. The suggested framework commences with tested & trained stages to guaranty precise infection identification. At the beginning, the input data is fetched from the CIC-MalMem-2022 (n.d.), CIC IoT dataset 2023 (n.d.), and CIC-IDS2017 (Intrusion Detection Evaluation Dataset n.d.) datasets and is processed in the preprocessing phase [8,9].

During this stage, extraneous fields are eliminated & incomplete datasets are imputed to enhance data accuracy, hence augmenting the model's theoretical learning efficacy. The interpolated information is now sent to the D2EBTN for conclusive judgment on malware identification. The suggested D2EBTN approach integrates stacking duality educational instruction, encompassing both progressive instruction and federation. In this context, the IL employs the Q model across several tasks to identify essential traits while also incorporating insights from prior tasks, hence mitigating severe forgetting. In a comparable manner, FL employs the Q modeling as several localized models alongside a singular universal modeling to derive significant insights without disclosing private information, thereby augmenting academic security while enhancing precise malware identification. Furthermore, the Q models integrates three Bi-LSTM models

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, September 2026
Impact Factor: 8.2

alongside a gated multidirectional concentration modules & a DBN modeling. The outcome identified is produced within the DBN framework. The use of a controlled multidirectional concentration modules aids the D2EBTN approach to enhancing vital characteristics & emphasizing essential data. Subsequently, the simulated model undergoes evaluation with examination information to identify precise viruses within the public cloud. Subsequently, the resultant procedure is assessed using academic metrics to verify its efficacy. This advanced methodology will enhance adaptation & comprehension, while also augmenting detecting precision through the integration of the double training paradigm. Figure 2 depicts the structure of the D2EBTN system.

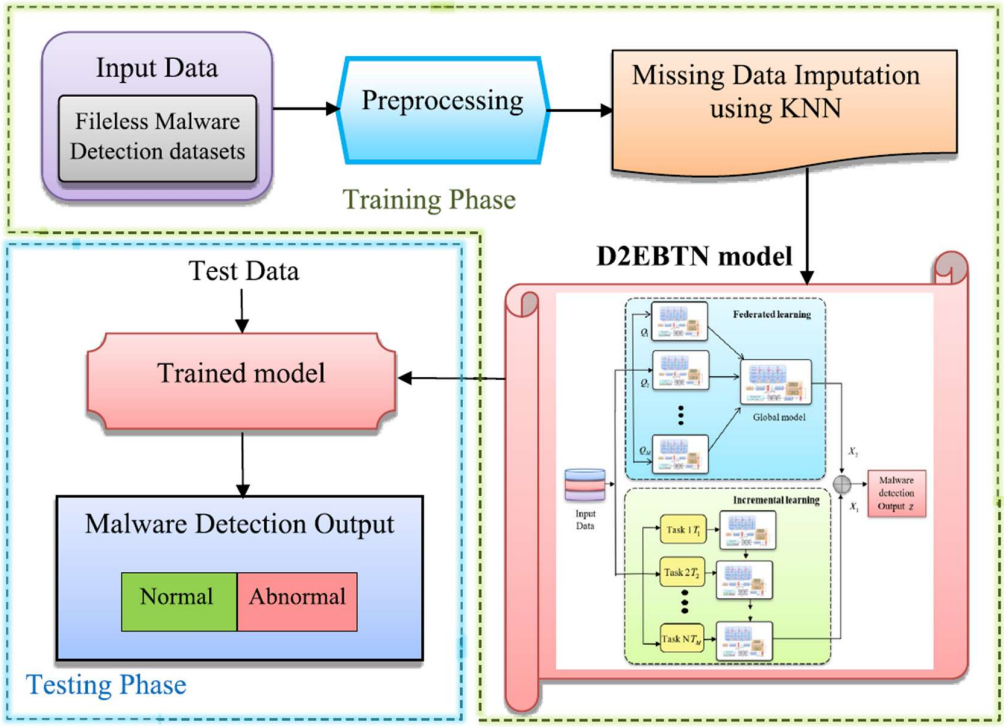


**Fig. 2 Structure of the suggested D2EBTN structure for cloud-based infection identification**

## IV. RESULT ANALYSIS

In the upcoming section, the comparative analysis of the D2EBTN model against GCRD (Satpathy and Swain 2025), CGOIDL-MD (Khalid et al. 2023), FL-DNN (Sherazi and Qureshi 2025), AutoML-CNN (Sihwail et al. 2018), Me-Mal-Det (Maniriho et al. 2024), and XG-Boost (Hossain and Islam 2024) methods using three publicly available datasets, CIC-MalMem-2022, CIC IoT dataset 2023, and CIC-IDS2017 are elucidated. To ensure an equitable contrast, current methodologies are evaluated using a technique akin to that of the suggested approach and assessed by analogous indicators. Consequently, this evaluation ensures a clear comprehension & importance of the D2EBTN concept.

As illustrated in Fig. 5.1, the D2EBTN system exhibits superior performance using the CIC-MalMem-2022 database when evaluated based on accuracy, F1-score, academic sensitivity, precision, & sensitivity parameters at 90% learning.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, September 2026
Impact Factor: 8.2

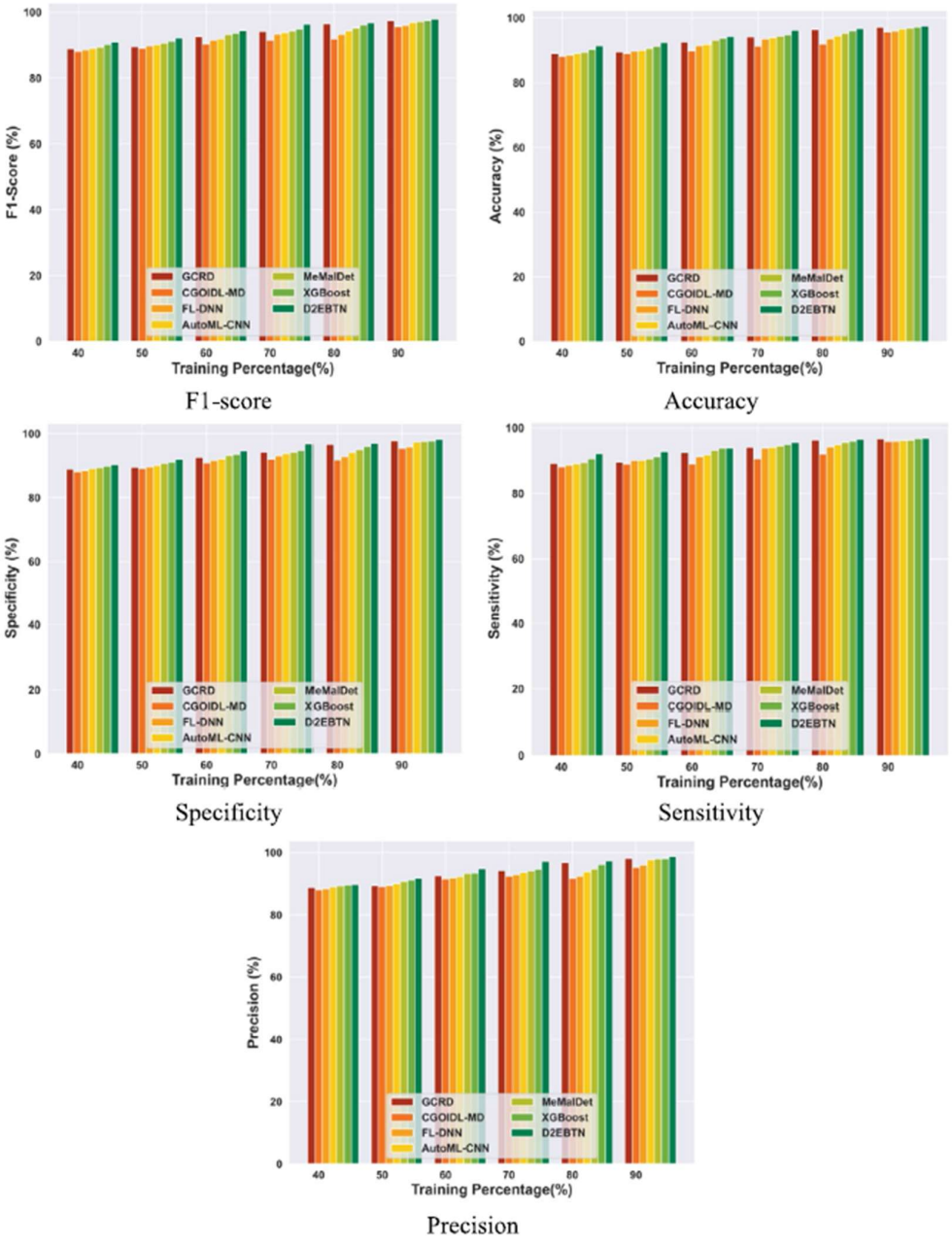


**Fig. 5.1 Comparison examination utilizing the CIC-MalMem-2022 database**

In comparison to existing techniques such as GCRD, CGOIDL-MD, & FL-DNN, the D2EBTN approach achieves notable enhancements in precision, including gains of 0.34%, 1.89%, & 1.63%. Conversely, the outcome of the AutoML-CNN approach is much inferior to that of the model suggested, resulting in a mere 0.93% enhancement in the F1-score measure. This examination demonstrates that the D2EBTN framework accounts for the periodic regularity of ransomware in cloud computing, hence attaining efficacy. Furthermore, the D2EBTN models achieves peak precision at an instruction percentage of 90%, demonstrating enhancements of 1.08%, 0.75%, and 0.68% over the AutoML-CNN, MeMal-Det, &

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, September 2026
Impact Factor: 8.2

XGBoost techniques. In a comparable manner, the model achieves enhancements of 1.03%, 0.99%, and 0.79% in academic sensitivity, while it realizes increases of 2.94%, 2.42%, and 1.01% in selectivity when compared to the CGOIDL-MD, FL-DNN, and AutoML-CNN methodologies at an instructional threshold of 90. Overall, the D2EBTN approach demonstrates encouraging outcomes as it utilizes the capabilities of an advanced gated multidirectional attention procedure & holds promise for immediate application.

## V. CONCLUSION

This study proposed the D2EBTN framework to improve malware and spyware detection in cloud environments by integrating incremental and federated learning with an attention-based deep learning architecture. The framework employs preprocessing techniques to remove redundant attributes and address incomplete data, followed by a hybrid learning architecture consisting of three Bi-LSTM models, a gated multidirectional attention mechanism, and a Deep Belief Network. The Bi-LSTM components enable the extraction of complex temporal dependencies, while the gated multidirectional attention mechanism emphasizes the most significant features for malware identification. The integration of Incremental Learning enables continuous adaptation to newly emerging attack patterns while preserving knowledge obtained from previous learning tasks, thereby reducing catastrophic forgetting. Similarly, Federated Learning supports privacy-preserving collaborative training by allowing local models to contribute to a global model without directly exchanging sensitive data. The proposed framework was evaluated using CIC-MalMem-2022, CIC IoT Dataset 2023, and CIC-IDS2017 and compared with established approaches, including GCRD, CGOIDL-MD, FL-DNN, AutoML-CNN, Me-Mal-Det, and XGBoost. The comparative results demonstrate the effectiveness of D2EBTN, particularly on the CIC-MalMem-2022 dataset at a 90% training level. The framework achieved improvements in precision of 0.34%, 1.89%, and 1.63% compared with GCRD, CGOIDL-MD, and FL-DNN, respectively. It also demonstrated improvements over AutoML-CNN, Me-Mal-Det, and XGBoost in precision, sensitivity, and selectivity. These results indicate that the combination of dual learning, temporal modelling, and attention-based feature prioritization can enhance malware detection performance while supporting adaptability and privacy. Nevertheless, further investigation is required using larger and more heterogeneous datasets, real-time cloud environments, and diverse emerging malware families. Future research can also investigate automated hyperparameter optimization, meta-heuristic feature selection, explainable AI, and lightweight deployment strategies to reduce computational overhead. Overall, D2EBTN provides a promising foundation for developing scalable, adaptive, and privacy-preserving intelligent malware detection systems for modern cloud environments.

## REFERENCES

1. Al Mamun A, Nath A, Dey SK, Nath PC, Rahman MM, Shorna JF, Anjum N (2025) Real-time malware detection in cloud infrastructures using convolutional neural networks: a deep learning framework for enhanced cybersecurity. Am J Eng Technol 7(03):252–261
2. Aljabri M, Alhaidari F, Albuainain A, Alrashidi S, Alansari J, Alqahtani W, Alshaya J (2024) Ransomware detection based on machine learning using memory features. Egypt Inform J 25:100445
3. Borana P, Sihag V, Choudhary G, Vardhan M, Singh P (2021) An assistive tool for fileless malware detection. In: 2021 World Automation Congress (WAC) (pp. 21–25). IEEE.
4. Aslan Ö, Ozkan-Okay M, Gupta D (2021) Intelligent behavior-based malware detection system on cloud computing environment. IEEE Access 7(9):83252–83271
5. Baawi SS, Oleiwi ZC, Al-Muqarm AM, Al-Shammary D, Sufi F (2025) Efficient malware detection based on machine learning for enhanced cloud privacy protection. Evolving Syst 16(1):30
6. CIC-MalMem-2022, n.d. CIC-MalMem-2022 dataset: (n.d.) https:// www. unb. ca/ cic/ datas ets/ malmem-2022. html. Accessed on December 2025.
7. CIC IoT dataset 2023 dataset: (n.d.) https:// www. unb. ca/ cic/ datas ets/iotda taset- 2023. html, Accessed on December 2025.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, September 2026
Impact Factor: 8.2

8. El Mouhtadi W, Maleh Y, Mounir S (2025) Machine learning techniques for enhanced malware detection in portable executable files. In: The Proceedings of the International Conference on Smart City Applications (pp. 784–796). Springer, Cham
9. Hossain MA, Islam MS (2024) Enhanced detection of obfuscated malware in memory dumps: a machine learning approach for advanced cybersecurity. Cybersecurity 7(1):16
10. Intrusion detection evaluation dataset (CIC-IDS2017), (n.d.) https:// www.unb. ca/cic/datasets/ ids- 2017. html, Accessed on December 2025
11. Ispahany J, Islam MR, Islam MZ, Khan MA (2024) Ransomware detection using machine learning: a review, research limitations and future directions. IEEE Access. https:// doi. org/ 10. 1109/ACCESS. 2024. 33979 21
12. Khalid O, Ullah S, Ahmad T, Saeed S, Alabbad DA, Aslam M, Buriro A, Ahmad R (2023) An insight into the machine-learning based fileless malware detection. Sensors 23(2):612
13. Khushali V (2020) A review on fileless malware analysis techniques. Int J Eng Res Technol (IJERT) 9(05)
14. Kim H, Kim M (2024) Malware detection and classification system based on CNN-BiLSTM. Electronics 13(13):2539
15. Kimmel JC, Mcdole AD, Abdelsalam M, Gupta M, Sandhu R (2021) Recurrent neural networks based online behavioural malware detection techniques for cloud infrastructure. IEEE Access 4(9):68066–68080
16. Kumar S, Dohare U, Kumar K, Dora DP, Qureshi KN, Kharel R (2018) Cybersecurity measures for geocasting in vehicular cyber physical system environments. IEEE Internet Things J 6(4):5916–5926
17. Liu J, Feng Y, Liu X, Zhao J, Liu Q (2023) MRm-dldet: a memory resident malware detection framework based on memory forensics and deep neural network. Cybersecurity 6(1):21
18. Maniriho P, Mahmood AN, Chowdhury MJM (2024) Me Mal Det: a memory analysis-based malware detection framework using deep autoencoders and stacked ensemble under temporal evaluations. Comput Secur 142:103864