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: A Review

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 adoption of cloud computing, mobile platforms, smart cities, and cyber-physical systems has increased the complexity and diversity of cyber threats, making intelligent attack and malware detection an essential component of modern cybersecurity. Machine learning (ML) and deep learning (DL) techniques have increasingly been investigated for detecting attacks such as Distributed Denial-of-Service (DDoS), Denial-of-Service (DoS), replay attacks, intrusions, anomalies, and malware. This review presents a comparative analysis of existing attack-detection approaches based on their objectives, learning paradigms, algorithms, detected attacks, reported accuracy, and future research directions. The reviewed studies demonstrate that conventional ML algorithms, including Decision Tree, Random Forest, Support Vector Machine, Naïve Bayes, K-Nearest Neighbour, Logistic Regression, and XGBoost, can achieve high detection accuracy under specific experimental conditions. For example, several studies reported accuracies above 97% for DDoS, intrusion, and malware detection. However, performance varies considerably depending on the attack type, dataset, feature representation, and experimental environment. The review further investigates malware feature-extraction strategies, particularly static, dynamic, and hybrid analysis. Static analysis utilizes information such as application manifests, permissions, API calls, opcodes, and code characteristics, whereas dynamic analysis considers system calls, network traffic, user interactions, and system-resource behaviour. Hybrid analysis combines these approaches to obtain more comprehensive behavioural information, although it generally requires greater computational resources and processing time. Existing research also indicates that different feature representations significantly influence classifier performance, with methods such as Gradient Boosting, Random Forest, Logistic Regression, and evolutionary optimization achieving promising results. Despite these advances, challenges remain in handling evolving attacks, large-scale datasets, computational overhead, feature variability, and the generalization of models across different environments. Therefore, future research should focus on adaptive and scalable ML/DL frameworks capable of integrating heterogeneous features and continuously learning from emerging threats.*

Keywords: Quick Response (QR) Codes; QR-Code Extracting; Adaptive Thresholding; Deep-Learning

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.

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

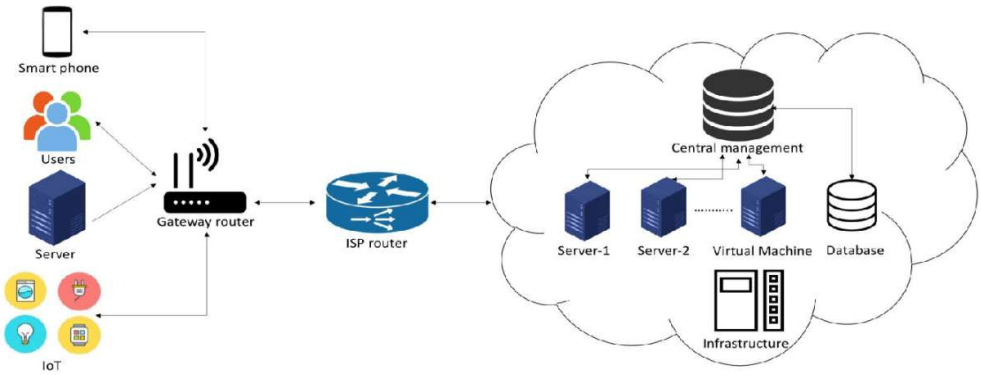

Figure 1: Cloud Computing (CC) Infrastructure

CC service models are represented in figure 1.2 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.

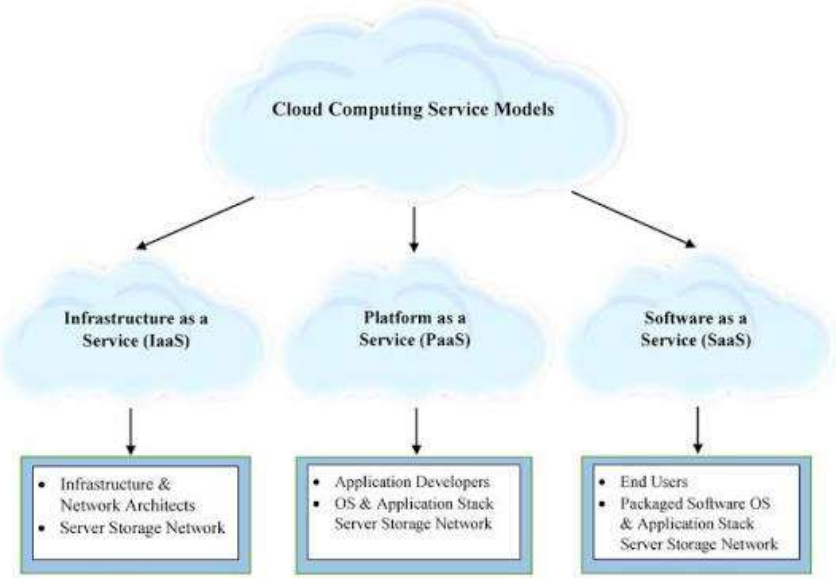

Figure 2: Cloud Computing (CC) Service Models

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. REVIEW BASED ON DIFFERENT ATTACKS

Comparisons of different attacks and different machine learning algorithms to detect these attacks have been shown in Table 1.

Table 1: Review based on different attacks

Authors	Objective	Algorithm used	Detected Attack	Technique	Accuracy	Future Scope
Marwane, Kathali et al.]	Protecting cloud providers from DDoS attacks	Supervised	DDoS	Naive Bayesian, C4.5 and K-Means	Accuracy: 91.4%,98.8 % and 95.9%	In order to identify and mitigate attacks posed by security challenges, the proposed framework will be developed into a prototype system for detection and mitigation of attack traffic.
Wani et al	Computing over cloud infrastructure & detecting DDoS attacks	Supervised	DDoS	Support Vector Machine, Naïve Bayes, and Random Forest	Accuracy: 99.7%, 98.0% and 97.6%	The future work will include a broader range of attack and selection techniques.
Nguyen Hoang et.al	Detecting and isolating mobile cloud systems is feasible before serious damage is caused by cyber threats.	Unsupervised	Cyber attack	NSL-KDD, UNSW-NB 15, and KDD cup	Accuracy: 90.99%, 95.84%, and 97.11%	Real-time testing will be performed on the experiment proposal on real devices.
Kushwah et al	A technique to detect DDoS assaults in CC	Unsupervised	DDoS	Voting extreme learning machine (V-ELM), a type of ANN.	Accuracy: 99.18%	Performance was also analyzed based on differences in system parameters.
Javadpour, Amir et al.,	By using intrusion detection systems and their various architectures, improve intrusion detection accuracy in cloud computing	Supervised	Intrusion	Decision tree, random forest, CART algorithm and neural network	Accuracy: 99.8%, 99.8%, 99.8% and 99.98%	Additionally, majority voting, which incorporates information into classifications, can improve them.
Valliam mal Shaju et. al	Cyber physical system defense through deep learning algorithms is discussed in detail.	-	Cyber attack	Distributed deep-learning	Accuracy: 99.27%	Contrasting newly acquired datasets using dispersed deep learning techniques such as SVM, decision trees, and additional neural networks

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

Nguyen Hung et al.	Maximizing the effectiveness of mitigation.	Supervised	DDoS	K-Nearest-Neighbor (KNN) and XG Boost	Accuracy: 98.0%	Contrasting newly acquired datasets using dispersed deep learning techniques such as SVM, decision trees, and additional neural networks
Elsaeidy, Asmaa et al., 2020	In smart cities, defending against retort attacks	Supervised	Replay attack	convolutional neural network (CNN)	Accuracy: 98.04%	There are not enough mitigating factors in these cases, thus further research is required in the near future.
Nur Hani et al,	Developed a defense system using three ML Algorithm	Supervised	Malware	K-Nearest Neighbors (K-NN), Decision Tree (DT) & Support Vector Machine (SVM)	Accuracy: 94% , 99% and 91%	Machine learning techniques should be included into security frameworks for smart cities.
J Pavithra et. al., 2020	Detecting malware and benign files from the dataset sample with accuracy	Supervised	Malware	Random forest (RF), Linear Regression (LR), and Decision Tree (DT)	Accuracy: 99.43%, 54% and 99%	Large datasets can be used to test algorithms.
Khilar et. al., 2019	Improvements are needed in cloud security	Supervised	DoS	KNN, Decision Tree, Logistic Regression, Naïve Bayes	Accuracy: 92.81%	The role-based accesscontrol paradigm should be used in conjunction with the suggested strategy to create a cloud environment that is more secure.
Bhamare et. al.,	The model is trained using supervised learning. Afterwards, test data from ISOT is used to validate the models.	Supervised	Anomaly Detection	Decision tree, Logistic Regression, SVM	Accuracy : 88.67%, 89.26% and 68.06%	Additional research is required in the domains of anomaly detection, clustering, and unsupervised machine learning.

In the January 2002 edition of Virus Bulletin journal [2], 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

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

findings are not impacted by the name issue. The authors of [3] 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.

III. REVIEW BASED ON FEATURE EXTRACTION METHODS

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.

IV. FEATURE EXTRACTION METHODS ASSOCIATED WITH HYBRID ANALYSIS

Machine learning also makes use of hybrid analysis. Using a model of system call analysis, ML techniques were applied in [11] to assess system calls and manifests. The machine learning model was trained with JSON files from the Androguard and Kaggle datasets. After testing and training the SVN, LR, kNN, and RF models, a JSON file was produced. LR was shown to be the most accurate (93%), of all the machine learning approaches currently in use. [12] explains how to use Manifest and System call analysis on the Drebin dataset to find malware. Subsequently, the data was categorized using ML algorithms based on RF and NB, with Application function decision procedures achieving the highest 96% accuracy.

In order to identify malware in And rototal datasets, the writers of [13] looked into Manifest and System Code Analysis. With an accuracy of 80%, the DT technique beat the NB approach in malware identification. In order to identify malware, Manifest and System Analysis were applied to the Drebin, Mal Genome dataset's API calls in [14]. The data was then classified using the methods NB, GB, kNN, SVM, MLP, DT, and RF. With an accuracy of 99.36%, it was found that GB had the highest accuracy. The methodology described in [15] offered a way to use machine learning techniques for the analysis of manifests and system calls. The Drebin, AMD, GP, and Github databases were used to collect the data.

Training and testing data have been utilized using the 97% TAN model. Algorithms were used in [16] for information flow analysis, code analysis, and system call analysis. The Google Play and Virus-total databases provided the information. Multi objective Evolution had the highest accuracy, at 95.15%, after training and testing using the Genetic algorithm. Malware is detected in [17] by using System Code Analysis and Manifest for API calls on the Drebin dataset.

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

Next, the classification was performed using the CBR, SVM, and DT algorithms. CBR was shown to have the highest accuracy, at 95%.

V. CONCLUSION

The reviewed literature demonstrates that machine learning and deep learning have become important approaches for detecting diverse cyber threats in cloud, mobile, smart-city, and cyber-physical environments. The comparative analysis shows that algorithms such as Random Forest, Decision Tree, SVM, Naïve Bayes, KNN, Logistic Regression, XGBoost, neural networks, and CNN-based models have achieved promising detection performance for DDoS, DoS, intrusion, replay attacks, anomalies, and malware. However, the reported performance cannot be considered universally applicable because it is strongly influenced by the dataset characteristics, attack category, feature-selection strategy, and experimental configuration. The review of malware detection further establishes that feature extraction is a critical stage of the detection process. Static analysis provides useful information from manifests, permissions, API calls, opcodes, and program code without executing the application, while dynamic analysis captures behavioural characteristics such as system calls, network traffic, user interactions, and resource utilization. Hybrid analysis combines static and dynamic characteristics and can provide more comprehensive representations of malicious behaviour, but its higher computational and time requirements remain important limitations. The reviewed studies also reveal several research gaps, including dependence on limited or outdated datasets, difficulty in detecting previously unseen and evolving attacks, high computational requirements, feature redundancy, and limited adaptability to changing attack behaviours. In addition, differences in malware naming conventions and feature representations can affect dataset consistency and classification outcomes. These limitations indicate the need for intelligent detection frameworks that can efficiently process heterogeneous and high-dimensional security data while maintaining scalability and robustness. Future research should therefore investigate adaptive learning, incremental learning, ensemble and deep-learning architectures, automated feature optimization, and explainable AI techniques for improved detection and decision-making. Such developments can contribute to the design of reliable, scalable, and continuously adaptive cybersecurity solutions capable of addressing emerging malware and attack patterns in dynamic 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.
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

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

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