

A Survey on Deep Learning Approaches for Malware Detection and Classification

Ms. Shivani Jain¹

¹ Assistant Professor, Department of Computer Science and Applications, Mandsaur, India

ABSTRACT: Malware is a serious threat in the cybersecurity area because of its dynamic nature, the variety of malware families, stealth, propagation and the capability of evading traditional security products. Therefore, proper malware detection and classification are crucial for detecting malicious software and for securing computer systems from unauthorized access and data stealing, and for disrupting systems. This study covers all the bases when it comes to deep learning approaches for malware detection and classification. It covers the principles, different forms of malware, how to detect deep learning malware, how to represent data, obtaining features, and applications. The traditional detection methods are described with their drawbacks, namely based on signature, behavioral and heuristic methods. The report also delves into the methodologies used by deep learning to classify malware, namely CNNs and Bidirectional Long Short-Term Memory (BiLSTM) networks. BiLSTM models excel at learning sequential features from code-or behavior-related data, whereas CNN-based representation learning approaches excel at learning spatial features from malware representations. Moreover, the various detection techniques (static, dynamic and hybrid) are discussed so that their role in malware analysis can be understood. The survey identifies the current challenges and gaps in research and emphasizes the need for strong, scalable and adaptive deep-learning models to combat new malware threats and enhance cybersecurity protection.

KEYWORDS: Malware Detection, Deep Learning, Convolutional Neural Networks, BiLSTM, Cybersecurity.

1. Introduction

A growing number of malware detection algorithms have begun to rely on data mining techniques and machine learning (ML) to identify harmful files in recent years. ML algorithms can learn to identify malicious and benign instances concealed inside a training set. Using these simple examples, malicious code may be distinguished from benign code [1]. As the number of people using the Internet continues to expand and more and more services move online, such banking and shopping, cybercriminals have more and more opportunities to commit cybercrimes, driving up the costs associated with system security measures [2]. The fast increase in the worldwide damage cost caused by cyber maliciousness has researchers very interested [3]. Cybersecurity Ventures' Official Annual Cybercrime Report estimates that this cost reached around \$6 trillion in 2021. The most significant danger to cybersecurity is typically malware, which includes a broad range of threats such as viruses, worms, Trojan horses, rootkits, and ransomware. Malware may steal important data or damage systems directly [4][5]. Furthermore, malicious software is the most common kind of attack on

computers, networks, or users that aims to steal sensitive data or inflict harm. A worrisome 22.9% increase in the number of harmful software instances in the past several years indicates a significant uptick in dangers faced by computer users [6].

A major game-changer in natural language processing (NLP) has been DL's capacity to automatically learn complicated patterns and representations from massive volumes of unstructured text input [7]. In many NLP applications, DL models, which leverage multi-layer artificial neural networks, have achieved remarkable results, often surpassing conventional methods and, in some cases, human performance. Malware is a prevalent and ever-changing cybersecurity risk, with more complex attack techniques and vectors aimed at both platforms [8][9]. The exponential increase of polymorphic and zero-day variants is not taken into consideration by signature/heuristic techniques. The advent of DL as a paradigm for solving these problems is encouraging since it improves the accuracy and versatility of automated pattern identification by learning complex representations from basic data. Recently, several DL techniques have been developed for malware detection,

which use architectures that obtain high detection accuracy through learning a hierarchy of spatial features during executable malware detection [10][11].

A. Structure of the paper

The paper structure is as follows: Section II introduces the basic concepts of malware, types, families, and detection techniques, Recent work on malware detection and classification using CNN and BiLSTM models, including static, dynamic, and hybrid techniques, is reviewed in Section III, and Section IV summarizes recent research and gaps. Section V concludes the paper and provides future directions.

2. Fundamentals of Malware

Malware basics cover the most basic information about malware, including its nature, forms, and how it functions. There are several varieties of malware, including viruses, worms, trojans, spyware, rootkits, and botnets. Its main function is to carry out illicit operations by taking advantage of system flaws. The nature, propagation, attack mechanisms and behaviors of malware are fundamental factors to consider when creating detection and classification techniques. It is characterized by stealth, persistence, quick propagation, code obfuscation, polymorphism and adaptability. Malware is constantly evolving and, because of this, automated detection and classification and advanced DL techniques are crucial to effective cybersecurity protection.

A. Malware Types and Families

A growing number of risks and threats come from various types of malware. Thus, malware classification is critical to help incident responders react rapidly. Currently, many families of malware are studied in research and each family includes various samples and types of malware. Intuitively, families of malware group malware that have similar characteristics into the same family and the characteristics can be recognized by patterns. In order to correctly categorize types of malware into malware families, which have similar characteristics [12].

- **Adware:** A kind of potentially unwanted program (PUP), "it" stands for "advertising-supported software," and it shows intrusive adverts to the user. Adware is software that displays adverts, usually in an annoying or bothersome way, in order to make money for its creators.

- **Vir Tool:** Malware detection prefixes let antivirus and anti-malware software identify potentially malicious software, utilities, and components. The term "Vir Tool" indicates that the detected item is not inherently malicious by itself, but it could be utilized or exploited by malicious software.

- **Spyware:** Malicious software aims to covertly track and record a user's actions on a device or computer. Spyware secretly collects personal information from users without their knowledge or permission. This data can contain personal information, login details, browsing habits, and other sensitive data.

- **Worm:** Self-replicating and disseminating independently on computer networks. Worms do not need an associated program to attach themselves to. Rather, they take advantage of weaknesses in network services or operating systems to proliferate across computers. Worms can congest networks, slow down the system's performance, and in some cases provide payloads with malicious intent.

- **Trojan:** A form of malicious software that is named for the Trojan horse. Unlike viruses and worms, Trojans multiply no more than they are prepared to be installed. Trojans are not contagious like worms or viruses, but they do wait to be implanted before reproducing. Trojans can then be activated to carry out multiple harmful activities in the background without user knowledge.

- **Backdoor :** A hidden or alternative route of access into a computer system, usually unauthorized. The main reason for a backdoor is to help someone gain unauthorized access to the system and gain control over it, often in a stealthy manner. Backdoor malware can be used in several different ways to attack, and can be a very serious security threat [13].

- **Password stealer:** It covertly steals sensitive login credentials, including usernames and passwords, from the compromised systems they infect. Malware of this kind often targets user accounts in an effort to steal sensitive information, commit financial fraud, or steal identity [14][15].

- **Rogue:** Malicious software that exhibits rogue or deceptive behavior, often with the intention of misleading or defrauding users [16].

B. Malware Detection and Classification Methods

There is a procedure called malware detection that must be implemented in order to locate and identify potentially harmful files [17]. Because of this, new malware detection technologies are always emerging; nevertheless, none of these methods have been shown to be 100% successful against all malware families or in all situations. Three approaches have been devised for the purpose of detecting malicious software: heuristic-based, signature-based, and behavioral-based. Two key features—signatures and behaviors—form the basis of these techniques.

1) Signature

Several studies have looked at using a comparison

between the freshly acquired file signature and an existing unique signature that has been either statically or dynamically modified as a way to improve malware detection and classification models [18]. Numerous formats are possible for signatures, such as an entropy value, a series of API calls, or a string of opcodes or bytecode. Synthesize the retrieved strings using frequency vectors to generate static string-based signatures capable of detecting VBasic malware. As an alternative, static signatures may be generated using binary vectors and n-grams. Malicious static signatures were created using data mining opcode statistics.

2) Behavioral

A model can be trained to identify known malicious behaviors and any similar ones in terms of false positives by first collecting exhibited behaviors while monitoring executable files in an isolated environment. Then, feature extraction techniques are used to extract sensitive features [19]. This method surpasses the signature-based approach due to its ability to detect both current and novel malware activities through the collection of runtime behaviors. Studies covered in the literature review aimed at boosting malware detection rates using behavioral-based techniques that utilized common, sequential, and continuous behaviors.

3) Heuristic

Heuristics have been used in several studies to strengthen models that may detect malicious intent. These heuristics are essentially broad principles that are used to assess the data that is collected through static or dynamic analysis [20]. Expert analysts' expertise and experience can be used to manually build the rules, or

automated methods like ML and the YARA tool can be used.

In addition, static signature-based methods for finding malware have been beaten when dealing with encrypted or hacked malware. Malware detection models in the literature are often built using API call data, but in the behavioral-based approach machine activity data is used to show how malware functions through continuous behaviors. In addition, while looking into malware clusters, writers usually rely on API calls and opcodes as their main data formats.

3. Deep Learning-Based Malware Detection and Classification

DL techniques are utilized by malware classification systems to classify malware families according to extracted characteristics and behavioral patterns. Utilizing the power of CNNs for malware analysis and BiLST networks for malware classification, the system is able to effectively process malware that is represented visually or geographically.

A. Malware Classification Based on CNNs

A new generation of image classification algorithms, CNNs, has demonstrated impressive feature learning capabilities. The transformation of binary executables into pictures has been the basis for several CNN-based methods for MC and MFC. Unlike higher dimensions CNNs-based, one dimension CNN (1D-CNN) can avoid loss from image inversion processes, and therefore, it can improve the accuracy of classification[21]. The proposed 1D-CNN for detecting malware was introduced by and the other proposed for classifying malware was proposed.

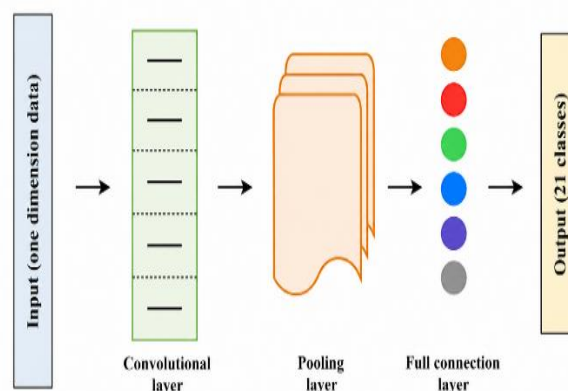


Fig. 1. 1D-CNN

The convolutional, pooling, and fully connected layers comprise a 1D-CNN, as shown in Fig. 1.

B. Malware Classification Based on BiLSTM

In LSTM, Hochreiter and Schmid Huber put forward the idea. It was employed to address the issues

associated with RNNs learning long-term dependencies. LSTM is ideal for sequential data due to its large memory capacity. The LSTM cell chain is a component of LSTM networks [22]. There are three gates in each LSTM cell: input, forget, and output. An LSTM cell's data input and control methods are defined by the "input/update" gate. What data may be erased from an LSTM cell is

determined by the forget gate. Which data is sent out of an LSTM cell is controlled by the output gate. Fig. 2

depicts the LSTM cell's architecture.

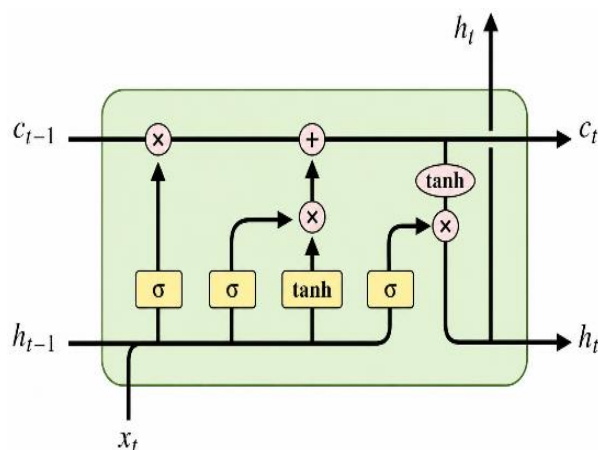


Fig. 2. Architecture of LSTM cell

The enhanced variant of LSTM is known as BiLSTM. By transmitting input sequences in both directions, BiLSTM is able to use information from both sides, unlike LSTM.

C. Applications of Deep Learning in Malware Detection

Dynamic detection, hybrid detection, and static detection are the three methods by which DL is employed in malware detection. Combining static and dynamic approaches, which assess malware both before and after execution, and in between, respectively, is what makes hybrid techniques so effective in detecting malware, as illustrated in Fig. 3.

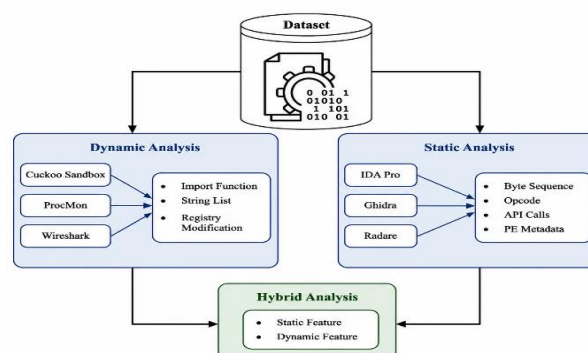


Fig. 3. Applications of deep learning in malware detection

1) Static Detection Method

Static detection processes a wide variety of data sources, including assembly and binary file formats, the functions that call them, and many more [23]. The software testing and development processes make heavy use of this method. Standard static detection techniques' underlying processes.

2) Dynamic Detection Method

In dynamic detection, samples of malware are run in sandboxes, emulators, or virtual computers in order to keep the environment under control. This process is used to observe the behavior of the sample when interacting with the system resources, with the purpose of identifying the patterns of behavior that are characteristic of malware [24]. This detection approach is commonly used throughout the operation and maintenance phase of software.

3) Hybrid Detection Method

There are substantial static analysis problems in detecting shelled malware. Dynamic analysis helps us understand how this code behaves in a simulated environment. This is a promising way to do it, but is not necessarily practical or scalable or secure. Therefore a hybrid detection method is developed to overcome the above limitations. This approach requires dynamic analysis of the malware shell, running the shell code, and then performing static analysis of the shelled code. This hybrid method combines the best of both worlds: dynamic and static. This detection approach is primarily employed during the software analysis stage.

4. Literature Review

The present survey explores the recent advances made in DL for detection and classification of malware including CNNs, RNNs, autoencoders and hybrid

models. It examines their effectiveness in recognizing complex malware patterns by using both static and dynamic attributes.

G. Abdiyeva-Aliyeva, et al. (2026) offer a DL method for classifying malware in critical infrastructure scenarios by combining network-level data with signs gathered during the virus's controlled execution. Present a curated set of sandboxed malware traces associated with high-impact threat categories to address the dearth of realistic and infrastructure-relevant datasets. Consistent features of these threats include interference with business operations, theft of credentials, and unwanted remote access [25].

V. S. P. Kumar, et al. (2026) offer a combined approach that uses both ML and DL models to categorize families of malware using text-based attributes derived from opcode sequences and code representation on a per-byte basis into many classes. Conduct experiments on the Microsoft Malware Classification dataset. PCA is used to reduce dimensionality, and TF-IDF and n-grams are used for feature extraction [26].

S. Akter and M. S. Islam (2025) examine a wide variety of methods, including basic classifiers (Random Forest, Decision Tree), gradient boosting techniques (XGBoost, Cat Boost), and several DL architectures (CNN, RNN, LSTM, GRU), as well as hybrid models (CNN-LSTM, GRU-LSTM). The Adam optimizer was used to optimize the models, and the categorical cross-entropy was used to construct the loss function. A validation loss-based early stopping mechanism was implemented to prevent overfitting [27].

J. Snead and K. Ghosh (2025) used clustering based on trace similarity calculation and malware trace sequence analysis with a biological twist. Everyone assumes that the malware traces pulled from the community detection methods are accurate representations of malware families. As a means of malware family prediction, the DL framework makes advantage of the most prevalent characteristics found in the retrieved traces. DL models implemented as ANNs are examined and the outcomes of experiments are detailed in this study [28].

P. M. Krishna, et al. (2024) apply methods for DL to find viruses in photos. Motivating this research was the

need for more efficient and accurate detection methods. Initially, need to gather both malicious and benign data pictures. looked at picture augmentation techniques to diversify the dataset. The next step was to use the processed dataset to train a CNN [29].

S. H. Yaddala, et al. (2024) apply a number of DL algorithms to a massive collection of malware samples in an effort to outperform conventional approaches in terms of detection speed and accuracy. The expected results are the development of a comprehensive, adaptable model that can effectively combat emerging threats of malware, thereby improving cybersecurity measures. The article discusses how DL is changing the game in the world of cybersecurity by transforming the way malware is classified. Some typical DL procedures that investigations include using CNNs for image-based classification and combining DL with traditional feature engineering techniques [30].

A. Prayoga, et al. (2023) analyze system calls, network flows, and static application properties to identify and categorize Android malware; assesses single-view DL systems as well as multi-view DL designs. To process system calls, the DL models use Bi-LSTM and Bi-GRU; to process network flows and static characteristics, they use MLP. In a single-view DL setup, the model handles each feature separately. Thanks to the concatenate function, multi-view DL allows a single model to take care of all three elements. Results from training and testing show that Bi-GRU and Bi-LSTM models perform well for binary classification architectures. They successfully achieve outcomes of up to 100%[31].

The majority of existing work is based on specific malware types, representations, or detection approaches, with few studies addressing a comprehensive framework. Most use benchmark datasets that do not accurately reflect the changing and zero-day malware. However, issues like dataset diversity, model interpretability, computational complexity, adversarial robustness, and generalization are yet to be addressed. Future studies should thus focus on scalable, explainable, multimodal, and adaptive DL-based systems that can effectively identify new malware types in a variety of real-world cybersecurity contexts. Table I summarizes recent research on DL approaches for malware detection and classification, including studies' methods, findings, limitations, and plans for further investigation.

TABLE I. SUMMARYRYSUMMERY OF THE STUDY ON DEEP LEARNING APPROACHES FOR MALWARE DETECTION AND CLASSIFICATION

Reference	Study On	Approach	Key Findings	Challenges	Future Direction
Abdiyeva-Aliyeva et al. (2026)	Malware classification for critical infrastructure	Deep learning using behavioral and network-level indicators from sandboxed malware traces	Developed an infrastructure-relevant dataset and classified high-impact malware threat categories	Limited realistic and infrastructure-specific datasets	Develop larger, diverse, and real-world critical infrastructure malware datasets

Kumar et al. (2026)	Multi-class malware family classification	Hybrid ML-DL model using opcode and byte-level features; TF-IDF, n-gram, and PCA	Effectively classified multiple malware families using text-based code representations	Feature complexity and dependence on benchmark datasets	Explore advanced hybrid models and more robust feature representations
Akter and Islam (2025)	Malware detection and classification	RF, DT, XGBoost, CatBoost, CNN, RNN, LSTM, GRU, and hybrid CNN-LSTM/GRU-LSTM	Compared traditional, deep learning, and hybrid models with optimization and early stopping	Risk of overfitting and model complexity	Improved generalization using adaptive and optimized hybrid architectures
Snead and Ghosh (2025)	Malware family identification using trace similarity	Biology-inspired malware trace analysis, clustering, and ANN-based prediction	Prototypical trace features supported malware family prediction	Dependence on trace quality and accurate family representation	Investigate advanced trace analysis and scalable deep learning methods
Krishna et al. (2024)	Image-based malware detection	CNN with malware and benign images and image augmentation	Demonstrated deep learning capability for detecting malware represented as images	Dataset diversity and image representation limitations	Develop robust image-based models using larger and more diverse datasets
Yaddala et al. (2024)	Deep learning-based malware classification	CNN-based image classification combined with traditional feature engineering	Highlighted improved accuracy and faster detection using deep learning	Adaptability to new malware and scalability challenges	Develop adaptive models capable of detecting emerging and unknown malware
Prayoga et al. (2023)	Android malware detection and classification	Single-view and multi-view DL using Bi-LSTM, Bi-GRU, and MLP on system calls, network flows, and static features	Multi-view learning integrated heterogeneous features, while Bi-GRU and Bi-LSTM achieved strong binary classification performance	Feature integration and model generalization	Explore multimodal, lightweight, and explainable models for evolving Android malware

5. Conclusion and Future Work

The ever-changing nature and high level of complexity of cyber threats have made malware detection and categorization crucial parts of contemporary cybersecurity. When it comes to unknown, disguised, and ever-changing malware, these signature-, behavior-, and heuristic-based approaches have their uses, but they fall short. Two excellent DL models that excel at learning complicated SPATIAL and SEQUENTIAL patterns from malware data are CNN and BiLSTM. Further, structural and behavioral properties are exploited to assist in thorough malware analysis with static, dynamic, and hybrid detection approaches. In summary, the potential of DL in automated malware identification and family classification is quite promising and it has the potential to minimize manual feature design. Issues such as malware change, data quality, model generalization, adversarial attacks and detection of new attacks that have never been seen are still important research problems. The proliferation of powerful, flexible and adaptive DL tools is therefore critical to strengthening cybersecurity defenses and making the most effective detection of malware in the changing threat environment.

Adaptive and scalable DL models for detecting zero-day and evolving malware should be the focus of future research. Key focus areas should be multimodal learning, explainable AI, strong datasets, adversarial resilience, and real-time detection to enhance generalization, explainability, and implementation in various cybersecurity contexts.

REFERENCES

1. A. Souri, M. Norouzi, and P. Asghari, "An analytical automated refinement approach for structural modeling large-scale codes using reverse engineering," *Int. J. Inf. Technol.*, vol. 9, no. 4, pp. 329–333, 2017, doi: 10.1007/s41870-017-0050-7.
2. S. R. Somula, "Intelligent Fraud Prevention: Implementing AML Solutions For Mcca Reimbursement Claims Protection," *Int. J. Comput. Eng. Technol.*, vol. 16, no. 1, pp. 3609–3622, Feb. 2025, doi: 10.34218/IJCET_16_01_249.
3. P. Kumar, "A Zero Trust-Based Approach to Modern Cybersecurity Challenges in Software Development," *Int. J. Emerg. Res. Eng. Technol.*, vol. 6, no. 9, pp. 113–122, September, 2025.

4. L. Caviglione et al., "Tight Arms Race: Overview of Current Malware Threats and Trends in Their Detection," *IEEE Access*, vol. 9, pp. 5371–5396, 2021, doi: 10.1109/ACCESS.2020.3048319.
5. S. P. Sivashanmugam, S. Kumara, A. Mohile, and D. R. Suram, "Improving Detection Accuracy of Phishing Attacks with Feature Selection and Machine Learning Techniques in Cybersecurity," in *2026 1st International Conference on Emerging Trends in Advancements and Applications of Computational Intelligence Techniques (ETA ACT)*, Bhubaneswar, India: IEEE, 2026, pp. 1–6, April. doi: 10.1109/ETA ACT69135.2026.11542138.
6. U. Urooj, B. A. S. Al-Rimy, A. Zainal, F. A. Ghaleb, and M. A. Rassam, "Ransomware Detection Using the Dynamic Analysis and Machine Learning: A Survey and Research Directions," *Appl. Sci.*, vol. 12, no. 1, 2022, doi: 10.3390/app12010172.
7. Y. LeCun and G. Hinton, "Deep Learning," *Nature*, vol. 521, pp. 436–444, 2015, doi: 10.1038/nature14539.
8. X. Zhang, K. Wu, Z. Chen, and C. Zhang, "Malcaps: A capsule network based model for the malware classification," *Processes*, vol. 9, no. 6, 2021, doi: 10.3390/pr9060929.
9. V. Koppireddy, D. G. Parasad, D. Patel, S. R. Somula, V. Kamaraj, and A. Meher, "AI Security Governance for Trustworthy Intelligent Systems: A Critical Analysis of Cyber Risks, Privacy Threats, and Responsible Deployment Frameworks," *Int. J. Res. Trends Innov.*, vol. 11, no. 07, pp. a407–a435, July, 2026, doi: 10.56975/ijrti.v11i7.214029.
10. T. Lu, Y. Du, L. Ouyang, Q. Chen, and X. Wang, "Android Malware Detection Based on a Hybrid Deep Learning Model," *Secur. Commun. Networks*, vol. 2020, pp. 1–11, 2020, doi: 10.1155/2020/8863617.
11. S. Chatterjee, "Advanced Malware Detection in Operational Technology: Signature-Based Vs. Behaviour-Based Approaches," *ESP J. Eng. Technol. Adv.*, vol. 1, no. 1, pp. 272–279, 2021, doi: 10.56472/25832646/JETA-V1I2P128.
12. M. Stamp, M. Alazab, and A. Shalaginov, *Malware Analysis Using Artificial Intelligence and Deep Learning*. 2021. doi: 10.1007/978-3-030-62582-5.
13. X. S. Nguyen and H. Nguyen, "A Malware Family Classification Approach Based on Deep Sequence Modeling," *J. Tech. Educ. Sci.*, vol. 20, no. 03, pp. 8–15, 2025, doi: 10.54644/jte.2025.1527.
14. V. K. R. K. Koppireddy, "Passwordless Authentication: Enhancing Security And User Experience In Modern Digital Ecosystems," *Int. J. Inf. Technol. Manag. Inf. Syst.*, vol. 16, no. 1, pp. 82–96, Feb. 2025, doi: 10.34218/IJITMIS_16_01_008.
15. S. Kakkar, B. Makkena, A. Chaudhary, and A. Kakkar, "Explainable Machine Learning Models for Credit Scoring for Risk Forecasting in Big Financial Datasets," in *2026 2nd International Conference on Cognitive Computing in Engineering, Communications, Sciences and Biomedical Health Informatics (IC3ECSBHI)*, Greater Noida, (Delhi NCR), India: IEEE, 2026, pp. 1384–1391, February. doi: 10.1109/IC3ECSBHI67834.2026.11469004.
16. N. K, I. K, S. S, N. V, and A. M, "Deep Learning Based Static Analysis of Malwares in Android Applications," in *Advances in Parallel Computing*, vol. 39, 2021. doi: 10.3233/APC210133.
17. F. A. Aboaoja, A. Zainal, F. A. Ghaleb, B. A. S. Al-rimy, T. A. E. Eisa, and A. A. H. Elnour, "Malware Detection Issues, Challenges, and Future Directions: A Survey," *Appl. Sci.*, vol. 12, no. 17, p. 8482, Aug. 2022, doi: 10.3390/app12178482.
18. D. Wael, A. Shosha, and S. G. Sayed, "Malicious VBScript detection algorithm based on data-mining techniques," in *2017 Intl Conf on Advanced Control Circuits Systems (ACCS) Systems & 2017 Intl Conf on New Paradigms in Electronics & Information Technology (PEIT)*, 2017, pp. 112–116. doi: 10.1109/ACCS-PEIT.2017.8303028.
19. A. Khalilian, A. Nourazar, M. Vahidi-Asl, and H. Haghighi, "G3MD: Mining frequent opcode sub-graphs for metamorphic malware detection of existing families," *Expert Syst. Appl.*, vol. 112, pp. 15–33, Dec. 2018, doi: 10.1016/j.eswa.2018.06.012.
20. R. Mosli, R. Li, B. Yuan, and Y. Pan, "Automated malware detection using artifacts in forensic memory images," in *2016 IEEE Symposium on Technologies for Homeland Security (HST)*, 2016, pp. 1–6. doi: 10.1109/THS.2016.7568881.
21. S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, "1D convolutional neural networks and applications: A survey," *Mech. Syst. Signal Process.*, vol. 151, p. 107398, 2021, doi: doi.org/10.1016/j.ymssp.2020.107398.
22. S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.
23. Y. Song, D. Zhang, J. Wang, Y. Wang, Y. Wang, and P. Ding, "Application of deep learning in malware detection: a review," *J. Big Data*, vol. 12, no. 1, p. 99, 2025, doi: 10.1186/s40537-025-01157-y.
24. A. Mohaisen, O. Alrawi, J. Park, J. Kim, D. Nyang,

- and M. Mohaisen, "Network-based Analysis and Classification of Malware using Behavioral Artifacts Ordering," *ICST Trans. Secur. Saf.*, vol. 5, no. 16, p. 156002, 2018, doi: 10.4108/eai.13-7-2018.156002.
25. G. Abdiyeva-Aliyeva, M. Balayev, and N. Ahmadli, "Deep Learning for Malware Classification in Critical Infrastructure Systems Using Behavioral and Network Indicators," in 2026 5th International Conference on Sentiment Analysis and Deep Learning (ICSADL), 2026, pp. 1078–1085. doi: 10.1109/ICSADL67539.2026.11451841.
26. V. S. P. Kumar, K. I. Koushik, K. Pradeep, B. G. V. S. Reddy, J. Avinash, and S. G. Pasha, "Optimized Machine and Deep Learning Approaches for Multi-Class Malware Classification from Text-based Features," in 2026 International Conference on Machine Learning and Autonomous Systems (ICMLAS), 2026, pp. 1027–1033. doi: 10.1109/ICMLAS67792.2026.11483801.
27. S. Akter and M. S. Islam, "Hybrid Deep Learning Architectures for Malware Detection: A Comparative Study," in 2025 IEEE International Conference on Telecommunications and Photonics (ICTP), 2025, pp. 285–289. doi: 10.1109/ICTP68765.2025.11415054.
28. J. Snead and K. Ghosh, "Deep Learning Models for Malware Family Detection," in 2025 13th International Symposium on Digital Forensics and Security (ISDFS), 2025, pp. 1–6. doi: 10.1109/ISDFS65363.2025.11011988.
29. P. M. Krishna, D. B. Kumar, and K. Ashwini, "Malware Detection in Image Data: A Deep Learning Approach," in 2024 5th International Conference on Communication, Computing & Industry 6.0 (C2I6), 2024, pp. 1–5. doi: 10.1109/C2I663243.2024.10894877.
30. S. H. Yaddala, M. P. C. Jothikumar, and A. Ashokkumar, "Malware Image Classification Using Deep Learning Approach," in 2024 First International Conference on Innovations in Communications, Electrical and Computer Engineering (ICICEC), 2024, pp. 1–5. doi: 10.1109/ICICEC62498.2024.10808890.
31. A. Prayoga, R. B. Hadiprakoso, R. N. Yasa, and Girinoto, "Deep Learning for Android Malware Detection and Classification Using Hybrid-Based Analysis: A Comparative Study," in 2023 IEEE International Conference on Cryptography, Informatics, and Cybersecurity (ICoCICs), 2023, pp. 309–313. doi: 10.1109/ICoCICs58778.2023.10277613.