Journal of Artificial Intelligence in Governance and Public Policy (JAIGPP)
Volume 1 | Issue 3 | July September-2026 | Pages 08-13
DOI: https://doi.org/xxxxxxxxxxxxx
e-ISSN: 3139-8693
Research Article OPEN ACCESS Available online: https://jaigpp.org/index.php/j

A Survey on Deep Learning Approaches for Malware Detection and Classification

Mr. Himanshu Baraiya

Department of Master Computer Application, Lakshmi Narnia College of Technology, Bhopal

Corresponding Author*Mr. Himanshu Baraiya (himanshub@lnct.ac.in, hemansu008@gmail.com)

Received: 15 July 2026 | **Revised:** 12 August, 2026 | **Accepted:** 03 September, 2026 |

***Abstract*—Malware comes in so many varieties and families, it can be extremely difficult to identify, it can be stealthy, it can propagate, and it can avoid traditional security solutions. It's a huge cybersecurity threat. So, malware detection (MD) and classification are essential components to detect malicious software and protect computer systems from unauthorized access, data stealing and disrupting systems. This survey provides a comprehensive review of deep learning techniques for malware detection and classification, including the fundamentals of malware, malware types and families, deep learning malware detection, data representation, feature extraction and applications. The limitations of the traditional detection techniques are discussed, namely signature-based, behavior-based, and heuristic-based approaches. Furthermore, the survey goes on to discuss the deep learning malware classification approaches, namely Convolutional Neural Networks (CNNs) and Bidirectional Long Short-Term Memory (BiLSTM) networks. CNN-based representation learning methods are better suited to learning spatial features from malware representations while BiLSTM models can learn sequential features from behavioral or code-based data. 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.*

I. INTRODUCTION

A growing number of malware detection algorithms have begun to rely on data mining techniques and 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. Malware, which includes viruses, worms, Trojan horses, rootkits, and ransomware, is considered the biggest threat to cybersecurity since it either takes crucial information or directly ruins systems [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].

In recent years, DLs capability to automatically learn complex patterns and representations from large amounts of unstructured text input has been a game-changer in the field of NLP [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 perhaps one of the most common and evolving cybersecurity threats and has a variety of attack surfaces, targeting a host of platforms [8][9][10]. Traditional detection methods like signature and heuristic are ineffective against fast-growing polymorphic and zero-day variants. Addressing these challenges, DL has been designed as an effective paradigm capable of learning complex representations directly from raw data, resulting in greater flexibility and PRE in automatic detection of malicious patterns. Malware detection using DL has recently seen a slew of new approaches, with architectures learning a hierarchy of spatial characteristics to achieve very accurate detections of executable malware [11][12].

A. Structure of the Paper

The paper is organized as follows: Section II introduces the basic concepts of malware, types, families, and detection techniques, Section III provides a comprehensive overview of current research on malware detection and classification using CNN and BiLSTM models, encompassing static, dynamic, and hybrid techniques. Section IV provides a concise summary of previous studies and identifies any gaps in the current literature. Section V wraps up the paper and points the way forward.

II. FUNDAMENTALS OF MALWARE

Malware basics cover the most basic information about malware, including its nature, forms, and how it functions. Malicious software, which includes computer programs like viruses, worms, Trojans, spyware, rootkits, and botnets,

Copyright © 2026 The Author(s).This is an open access article distributed under the terms of the Creative Commons Attribution 4.0 International (CC BY 4.0) License (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, adaptation, and reproduction in any medium, provided the original author(s) and the source are properly credited.

DOI: https://doi.org/xxxxxxxxxxxx

exploits security holes in a system to execute malicious actions. 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 [13].

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 [14].

- **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. The purpose of spyware is to gather sensitive data without the user's knowledge or consent. 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 [15]. 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 do not reproduce as viruses and worms do, but are waiting to be installed. 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 [16].
- **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 [17][18].
- **Rogue:** Malicious software that exhibits rogue or deceptive behavior, often with the intention of misleading or defrauding users [19].

B. Malware detection and classification methods

The software that has to be put in place to find and identify the files that are dangerous is known as the malware detection process [20]. This has resulted in the ongoing development of several malware detection technologies; nevertheless, no solution has hit a 100% success rate across all malware families and in all settings just yet. Accordingly, three methods for detecting malware have been devised, relying on signatures and behaviors, respectively: behavioral-based, heuristic-based, and signature-based.

1) Signature

A lot of studies have looked at ways to improve malware detection and classification models by comparing the newly acquired file signature with a previously stored unique signature that has been produced statically or dynamically [21]. A collection of API calls, a sequence of opcodes or byte codes, and an entropy amount are all examples of signatures. There are two main types of static signatures that have been established for identifying VBasic malware: one that uses frequency vectors to represent the extracted strings, and another that uses n-grams and binary vectors. By analyzing the statistical values of opcodes, harmful static signatures were also created.

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 [22]. This technique is superior to the signature-based approach as it gathers malware behaviors during runtime, making it able to identify both known and unknown malware behaviors. Most of the research in the literature study focused on behavioral-based strategies that employed common, sequential, and continuous behaviors to increase malware detection rates.

3) Heuristic

A number of studies have relied on heuristics to bolster their models for identifying malevolent intent by creating general rules to examine the retrieved data provided by either dynamic or static analysis [23]. 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 encrypted or hacked malware was being dealt with. Machine activity data, a component of the behavioral-based method, is often used to demonstrate malware functioning through continuous behaviors in the literature. API call data, on the other hand, is typically utilized to construct malware detection models through sequential behaviors. Data formats most commonly used by authors to examine malware clusters include API calls and opcodes.

Journal of Artificial Intelligence in Governance and Public Policy, 1 (3):08-13
DOI: https://doi.org/xxxxxxxxxxxx

III. 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. They are able to handle malware shown as images or geographical information by merging the capabilities of CNNs for malware analysis with BiLSTM networks for malware categorization.

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 ACC of classification [24]. The proposed 1D-CNN for detecting malware was introduced, and the other proposed for classifying malware was introduced.

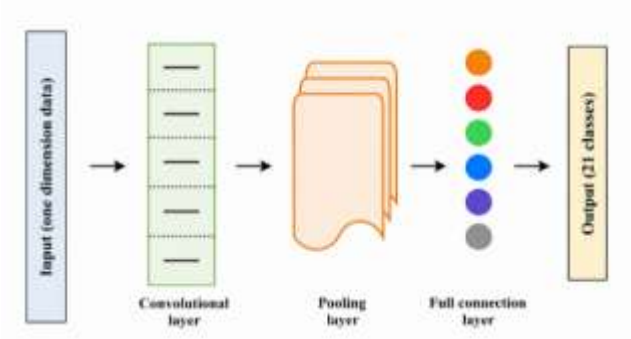

Fig. 1. 1D-CNN

The three layers that make up a 1D-CNN are the convolutional, pooling, and fully connected layers, as illustrated in Figure 1.

B. Malware Classification Based on BiLSTM

In LSTM, Hochreiter and Schmidhuber 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 [25]. Literally LSTM cells consist of three gates: input, forget, and output. What data is added to an LSTM cell and how it is controlled is decided 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. Figure 2 depicts the LSTM cell's architecture.

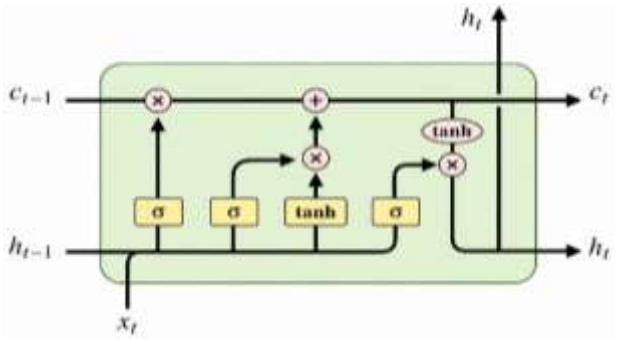

Fig. 2. Architecture of LSTM Cell

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

Malware detection makes use of DL in three ways: static detection, dynamic detection, and hybrid detection, as illustrated in Figure 3 To enhance the efficacy of malware detection, hybrid techniques combine static and dynamic approaches, which evaluate malware without execution and during execution, respectively.

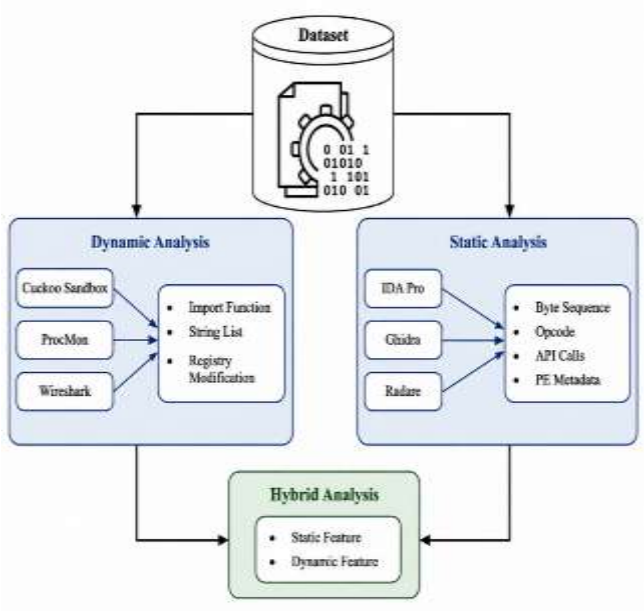

Fig. 3. Applications of Deep Learning in Malware Detection

1) Static detection method

Static detection examines a wide variety of data sources, including assembly and binary file formats, the functions that call them, and more [26]. 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 [27]. This detection approach is commonly used throughout the operation and maintenance phase of software.

3) Hybrid detection method

There are substantial static analysis problems to detecting shelled malware. The dynamics of this type of code in a virtual environment can be better understood with the help of dynamic analysis. This is a promising way to do it, but is not necessarily practical or scalable or secure. Consequently, in order to circumvent these restrictions, a hybrid detection approach is created. This method involves analyzing the malware shell dynamically, executing the shell code, and then analyzing the shelled code statically. This approach is a blend of both dynamic and static, combining the best attributes of each. This detection method is mainly applied during the software analysis step.

DOI: https://doi.org/xxxxxxxxxxxx

IV. Literature Review

The survey discusses the recent progress on DL for malware detection and classification, including CNN, RNN, autoencoder, and hybrid approaches. It analyzes their efficacy in recognition of intricate malware patterns based on static and dynamic characteristics.

G. Abdiyeva-Aliyeva, et al. (2026) provide a DL-based framework for malware classification in critical infrastructure situations, which integrates data collected at the network level with indicators collected during the malware's controlled execution. Propose a curated dataset of sandboxed malware traces linked to high-impact threat categories, such as industrial disruption, credential theft, and remote access activities, to address the scarcity of realistic and infrastructure-relevant datasets [28].

V. S. P. Kumar, et al. (2026) suggest a combined approach using both ML and DL models to categorize groups of malware according to textual characteristics derived from opcode sequences and code representations on a per-byte basis. The Microsoft Malware Classification dataset is used as an experimental dataset. TF-IDF and n-grams are used as feature extraction techniques, and PCA is used as a method of dimensionality reduction [29].

S. Akter and M. S. Islam (2025) examine a wide variety of methods, including basic classifiers (RF, DT), gradient boosting techniques (XGBoost, CatBoost), 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 derive the loss function. In order to avoid overfitting, a validation loss-based early stopping mechanism was put into place [30].

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 [31].

P. M. Krishna, et al. (2024) apply methods for DL to find viruses in photos. The necessity for better and faster detection methods prompted this research. Making a database containing both malicious and benign picture data is the initial step. Picture augmentation techniques to diversify the dataset. Then, a CNN was designed and trained on the processed dataset [32].

S. H. Yaddala, et al. (2024) utilize multiple DL algorithms on a large corpus of malware samples with a view to achieving greater ACC and speed of detection than traditional methods. 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 delves into the revolutionary way DL is reshaping malware classification in the world of cybersecurity. The use of CNNs for picture-based classification and the cooperation between DL and conventional feature engineering methods are among the common DL methodologies that explore [33].

A. Prayoga, et al. (2023) classify Android malware using system calls, network flows, and static application attributes, and evaluate single-view DL systems and multi-view DL architectures. For system call processing, the DL models use Bi-LSTM and Bi-GRU; for network flow and static feature processing, they employ MLP. The model in the single-view DL architecture deals with each feature separately. When dealing with all three aspects, multi-view DL employs a single model that uses the concatenate function. Based on training and evaluation results, Bi-GRU and Bi-LSTM models are the best for binary classification architecture, achieving results of up to 100% [34].

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 provides a summary of the recent studies related to DL techniques for malware detection and classification, detailing their methodologies, results, weaknesses, and future work.

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	Improve 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

DOI: https://doi.org/xxxxxxxxxxxx

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

V. Conclusion and Future Work

The ever-changing nature and growing complexity of cyber threats have made malware detection and categorization crucial parts of contemporary cybersecurity. These signature, behavior and heuristic-based techniques have their place but are not completely effective against unknown, obfuscated and changing malware. CNN and BiLSTM models are effective models in DL that offer great learning ability for complex spatial and sequential patterns in malware data. Furthermore, structural and behavioral characteristics are exploited for static, dynamic and hybrid detection to provide comprehensive malware analysis. 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. There are still challenges to dealing with changing malware, data quality, generalization of models, adversarial attacks, detection of new threats that haven't been seen before. The continued development of robust, flexible and adaptive DL tools is therefore crucial to strengthen cybersecurity defenses and successfully detect malware in an ever-changing environment of threats.

Future work should be directed towards creating adaptive and scalable DL models that will be able to identify zero-day and evolving malware. Multimodal learning, explainable AI, robust datasets, adversarial resilience, and real-time detection are key focus areas to improve generalization, explainability, and implementation across the range of cybersecurity applications.

Funding: This research received no external funding.
Institutional Review Board Statement: Not applicable.
Informed Consent Statement: Not applicable.
Data Availability Statement: The data supporting the findings of this study is available from the corresponding author upon reasonable request.
Conflicts of Interest: The author declares no conflict of interest

How to cite this article
H. Baraiya, "A survey on deep learning approaches for malware detection and classification," *Journal of Artificial Intelligence in Governance and Public Policy*, vol. 1, no. 3, pp. 10–13, July–September 2026., Doi.XXXXX

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 (ETAACT)*, Bhubaneswar, India: IEEE, Apr. 2026, pp. 1–6. doi: 10.1109/ETAACT69135.2026.11542138.

[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] K. Jangiti, "Design and Validation of a Machine Identity Governance Framework for AI Agents in Multi-Cloud Environments," in *SoutheastCon 2026*, Huntsville, AL, USA: IEEE, Feb. 2026, pp. 1–6. doi: 10.1109/SoutheastCon63549.2026.11476363.

[11] 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.

[12] 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.

[13] L. A. Yeruva, D. Singh, S. Suddala, N. Bhatt, and R. Uddin, "Augmented Data Management for Cache Performance, Cybersecurity, and Mobile Integration," *J. Comput. Mech. Manag.*, vol. 5, no. 3, pp. 280–293, june, Jun. 2026, doi: 10.57159/jcmm.5.3.26691.

[14] M. Stamp, M. Alazab, and A. Shalaginov, *Malware Analysis Using Artificial Intelligence and Deep Learning*. Cham: Springer International Publishing, 2021. doi: 10.1007/978-3-030-62582-5.

[15] N. Adik, "The Rise of Open and Free Networks: A Community-Driven Paradigm for Decentralized Connectivity," in *2025 IEEE First International Conference on Innovations in Engineering and Next-Generation Technologies for Sustainability (ICINVENTS)*, Coimbatore, India: IEEE, 2025, pp. 1–9, November. doi: 10.1109/ICINVENTS64613.2025.11402224.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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)*, IEEE, May 2016, pp. 1–6. doi: 10.1109/THS.2016.7568881.

[24] 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, Apr. 2021, doi: 10.1016/j.ymssp.2020.107398.

[25] 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.

[26] 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.

[27] 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.

[28] 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)*, IEEE, Feb. 2026, pp. 1078–1085. doi: 10.1109/ICSADL67539.2026.11451841.

[29] 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)*, IEEE, Mar. 2026, pp. 1027–1033. doi: 10.1109/ICMLAS67792.2026.11483801.

[30] 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.

[31] J. Snead and K. Ghosh, "Deep Learning Models for Malware Family Detection," in *2025 13th International Symposium on Digital Forensics and Security (ISDFS)*, IEEE, Apr. 2025, pp. 1–6. doi: 10.1109/ISDFS65363.2025.11011988.

[32] 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)*, IEEE, Dec. 2024, pp. 1–5. doi: 10.1109/C2I663243.2024.10894877.

[33] 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)*, IEEE, Oct. 2024, pp. 1–5. doi: 10.1109/ICICEC62498.2024.10808890.

[34] 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)*, IEEE, Aug. 2023, pp. 309–313. doi: 10.1109/ICoCICs58778.2023.10277613.