Journal of Theoretical and Applied Information Technology
29th February 2024. Vol.102. No 4

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

DEEP LEARNING FOR MALWARE DETECTION: LITERATURE REVIEW

JAWHARA BOODAI[1], AMINAH ALQAHTANI[2], AND KHALED RIAD[3,4]

[12]College of Computer Science and Information Technology, King Faisal University, Al-Ahsa 31982, Saudi Arabia

[3]Computer Science Department, College of Computer Sciences & Information Technology, King Faisal University, Al-Ahsa 31982, Saudi Arabia

[4]Mathematics Department, Faculty of Science, Zagazig University, Zagazig 44519, Egypt

E-mail: [1] 222452718@student.kfu.edu.sa, [2] 223002599@student.kfu.edu.sa, [3]kriad@kfu.edu.sa, [4]khaled.riad@science.zu.edu.eg

ABSTRACT

Malware is among the biggest cybersecurity threats, that are changing all the time to dodge traditional signature-based detection. In particular, machine learning, especially deep learning, is a promising method for malware detection. This paper provides an SLR of deep learning approaches for malware detection on Windows, Android, IoT, and other platforms. In all, we searched five major digital libraries and found 107 highly relevant studies published in 2015-2023. The SLR methodology consisted of well-formulated search queries, inclusion/exclusion criteria, and stringent full-text evaluation. Convolutional neural networks (CNNs) are most popular, learning spatial patterns from raw binaries. Malware sequential behaviors are modeled using LSTM networks. Spatial and temporal learning are combined in ensemble models such as CNN-LSTM which achieve high accuracy. But essential challenges persist, such as the generalization problem under obfuscation, lack of transparency, and lack of labeled real-world data. Although deep learning makes the malware detection more accurate than traditional methods, evasion attacks, interpretability, and data limitations need to be addressed. This SLR offers important insights into the strengths, tendencies, datasets, and weaknesses of deep learning for strong malware defense. With persistent threats, the use of effective AI-based approaches will only further grow in importance.

Keywords: *Deep Learning; Malware Detection; Convolutional Neural Networks; Long Short-Term Memory Networks*

1. INTRODUCTION

Deep learning involves learning multi-level data representations, with higher levels representing more abstract concepts. This enables deep learning models to learn highly complex functions directly from raw data without extensive feature engineering. Deep learning has proven very effective for uncovering patterns in high-dimensional data and is now applied across many domains. However, traditional machine learning often struggles to process raw, complex data. Malware refers to malicious code like viruses, Trojans, spyware designed to infect or damage computer systems. Malware developers use techniques like obfuscation to avoid detection by signature-based antivirus tools that rely on static pattern matching. However, malware variants frequently share common underlying behaviors that may potentially be detected using machine learning methods even when the code looks different. This makes deep learning promising for malware detection as it can potentially learn more sophisticated features compared to classic machine learning approaches. While research interest in leveraging deep learning for malware detection has surged in recent years, most published studies tackle only a specific malware platform, operating system, or variant. Comprehensive perspectives encompassing the full landscape are still lacking. To help address this gap, we conducted a systematic literature review (SLR) of research on deep learning techniques applied for malware and intrusion detection published from 2015-2023.

Although the research interest in the use of deep learning for malware detection has increased significantly over the past few years, most of the studies published focus on a particular malware platform, operating system, or variant. However, broad views that include the whole landscape are still missing.

Several surveys concentrate on malware detection for particular platforms such as Android [10] or IoT devices [11] Other studies have analyzed specific deep learning algorithms such as RNNs [12] or CNNs [13] for malware. Nevertheless, there is no comprehensive systematic analysis of the capabilities, datasets, limitations, and open problems for deep learning techniques and computing platforms to date.

In order to contribute to filling this gap, we performed an SLR of deep learning techniques used for malware and intrusion detection from 2015-2023. Our SLR methodology comprehensively reviews 107 highly relevant studies in order to provide a comprehensive overview for Windows, Linux, Android, IoT, and other platforms.

We thoroughly analyze the comparative performance of convolutional neural networks, recurrent networks, deep belief networks, autoencoders, and ensemble models for malware detection. Systematically, the trends, datasets, limitations, and future research directions are identified. This broader overview of the malware detection landscape has not been provided in previous surveys.

This review serves as a helpful guide for researchers who seek to develop better reliable deep learning-based malware defense systems by outlining the current state-of-the-art and open problems. As malware threats continue to expand and grow, the importance of AI-powered protection will only continue to rise.

1.1 Motivation

- ✓ Malware poses one of the biggest threats to computing platforms like personal computers, mobile devices, and the Internet of Things (IoT).
- ✓ As malware continues to increase in sophistication, traditional signature-based antivirus solutions are becoming inadequate.
- ✓ Machine learning, especially deep learning, has emerged as a promising approach for robust and generalizable malware detection.
- ✓ Deep learning models like convolutional neural networks (CNNs), recurrent neural networks (RNNs), and deep belief networks (DBNs) can automatically learn complex features and patterns from raw malware samples to detect new threats.
- ✓ This literature review systematically surveys recent research on deep learning techniques for malware detection across Windows, Android, IoT, and other platforms. By thoroughly analyzing trends, algorithms, datasets, limitations, and capabilities, it provides valuable insights to guide future research on AI-powered malware defense systems.
- ✓ As malware attacks persist and evolve, effective deep-learning solutions will only grow in importance.

The rest of the paper is organized as follows. Section 2 discusses the systematic literature review methodology followed to identify and analyze the most relevant studies. Section 3 provides a comprehensive review of deep learning techniques applied for malware detection across Windows, Android, IoT, Linux, and other platforms. For each platform, key algorithms, methods, datasets, and capabilities are analyzed. Section 4 presents a discussion of the major findings, including the predominance of CNNs, comparative effectiveness over machine learning, and limitations faced. Finally, Section 5 concludes with a summary of insights gained and implications for future research directions in this critical domain of malware detection using deep learning and AI.

1.2 Research Questions

Through this SLR, we aimed to thoroughly analyze the scope, trends, specific techniques and methods used, algorithms applied, challenges faced, and ability to generalize across the field. We sought to answer several key research questions:

- ✓ RQ1: Which computing platforms are most heavily targeted and impacted by malware attacks and threats?
- ✓ RQ2: What are the hot eras and trends in malware detection research, which platforms see the most focus, and which publication venues are most prominent?
- ✓ RQ3: What particular methods and techniques do researchers employ in order to detect malware using deep learning?
- ✓ RQ4: Which significant machine learning and deep learning algorithms have been used in order to detect malware?
- ✓ RQ5: What are the primary obstacles and restrictions of applying deep learning for efficient detection of malware and intrusions?
- ✓ RQ6: Are studies and suggested deep learning techniques useful for detecting Android malware encourage important characteristics

including adaptability, sustainability, and automated selection of of optimal algorithms?

- ✓ RQ7: Do Android-based malware detection methods proposed demonstrate ability to identify new, unknown malware variants, and what feature analysis techniques are used?
- ✓ RQ8: What datasets are most widely used and standard for evaluation of malware detection systems focused on Android and Windows platforms?

We systematically searched for and analyzed the most relevant studies on deep learning techniques for malware detection published from 2015 through 2023. Established rigorous guidelines for performing systematic literature reviews were carefully followed to obtain comprehensive insights without bias.

The results provide a thorough overview of trends, techniques, algorithms, datasets, limitations, open challenges, and future directions in this quickly evolving field. By shedding light on the current malware detection research landscape, this review serves as a valuable reference for researchers or engineers aiming to advance reliable deep learning-driven malware defense systems. In Figure 1, a flow chart shows the deep learning algorithms for malware detection. As malware threats persist and grow, effective AI-powered protection will only increase in critical importance. The training and testing techniques are used.

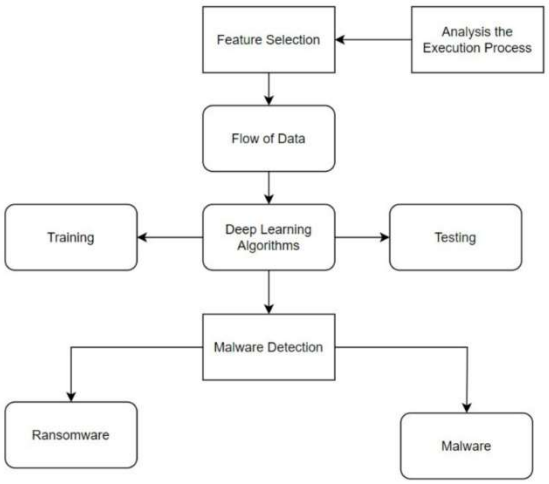

Figure 1: Flowchart of Deep learning algorithms for malware detection.

Malware detection can be formulated as a multi-class classification problem in machine learning, with the goal of categorizing files or applications into benign or one of several malware types. In Figure 2, a flow chart shows that the first step is extracting informative features like system calls, API calls, opcodes, string signatures, metadata, etc., from the programs. These features are then used to train classifiers like neural networks, SVMs, random forests, etc., on labeled benign and malware samples. The trained model based on classification and learning can classify new unseen programs into classes like adware, spyware, ransomware, trojans, worms, and viruses based on the extracted features. Using techniques like one-vs-all for multi-class classification, the model outputs predicted probabilities for each malware type. The program is assigned the class with the highest probability. Careful feature engineering and model tuning are critical for accurately detecting and categorizing the wide range of modern malware variants.

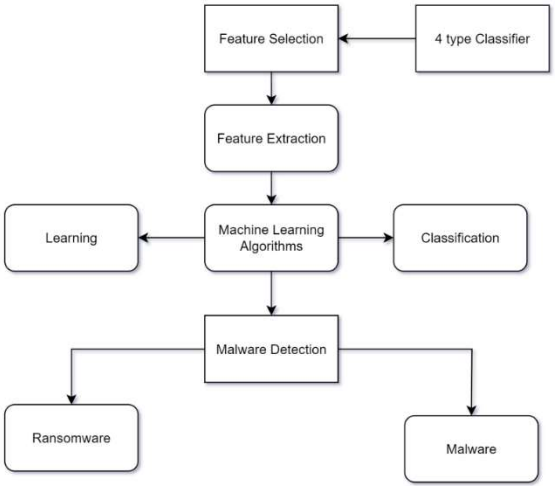

Figure 2: Flowchart of Machine learning algorithms for malware detection.

Below, Table 1 is the abbreviation table:

Table 1: Abbreviation Table.

Abbreviation	Definition
ML	Machine Learning
DL	Deep Learning
IoT	Internet of Things
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
DBN	Deep Belief Network
LSTM	Long Short-Term Memory
Bi-GRU-CNN	Bidirectional Gated Recurrent Unit - CNN
BiLSTM	Bidirectional LSTM
AMD	Android Malware Dataset
XGBoost	eXtreme Gradient Boosting
D.T	Decision Tree
R.F	Random Forest
API	Application Programming Interface
CPU	Central Processing Unit

ITMF	Image Texture Median Filtering
URL	Uniform Resource Locator
TFIDF	Term Frequency–Inverse Document Frequency
KNN	K-Nearest Neighbors
SVM	Support Vector Machine

1.3 Research Objectives

- ✓ To conduct a systematic literature review (SLR) surveying the application of deep learning techniques for malware detection across Windows, Android, IoT, and other platforms.
- ✓ To analyze the capabilities, algorithms, datasets, trends, limitations, and open challenges of using deep neural networks to detect malware based on the existing literature.
- ✓ To provide a comprehensive overview of the malware detection research landscape to guide future work on applying deep learning and AI for robust malware defense.
- ✓ To specifically examine the use of convolutional neural networks (CNNs), recurrent neural networks (RNNs), long short-term memory (LSTMs), deep belief networks (DBNs), autoencoders, and ensemble models.
- ✓ To assess the effectiveness of deep learning for malware detection compared to traditional machine learning approaches relying on manual feature extraction.
- ✓ To identify key obstacles faced in real-world deployment of deep learning-based malware defense systems.

This study focuses solely on reviewing existing literature and does not involve any novel data collection or experiments. The scope is limited to studies applying deep learning or machine learning techniques for malware detection. Broader cybersecurity topics like network intrusion detection or spam filtering are not included. The aim is to synthesize insights from prior research to inform future work on using AI to counter evolving malware threats.

1.4 Problem Selection

Malware was selected as the problem domain because it remains one of the most significant and evolving cybersecurity threats. Deep learning has emerged as a promising approach for robust malware detection that can automatically learn complex features from raw data. However, a comprehensive overview of deep learning techniques applied for malware defense across platforms was lacking.

2. RESEARCH METHODS

2.1 Search Strategy

Manually searching individual libraries for keywords related to deep learning and malware detection is inefficient. A better approach is to develop a comprehensive search query combining relevant keywords, synonyms, and abbreviations using logical operators like "OR" and "AND". For example, "Convolutional Neural Networks" and "CNN" should be combined with "OR" since they refer to the same technique. Similarly, we want studies discussing both "Deep Learning" and "Malware Detection", so these terms need "AND" linkage. Deep Learning, Deep Learning techniques, and malware detection were among the terms we used in our search. ("Deep Learning" OR "Convolutional neural network" OR "Deep belief network" OR "recurrent neural network" OR "CNN" OR "RNN" OR "DBN" OR "LSTM") AND ("malware") AND ("detection" OR "detect" OR "identification" OR "identify" OR "classification"). This was the final search query. This consolidated query enables a thorough yet efficient search for pertinent studies across keywords and terminology related to deep learning and malware detection.

2.2 Literature Screening Criteria

Searched 5 major digital libraries: IEEE Xplore, ACM DL, ScienceDirect, SpringerLink, Google Scholar.

Initial search results filtered using inclusion criteria:

- Peer-reviewed journal or conference papers
- Published between 2015-2023
- English language
- Full text available

Exclusion criteria to remove irrelevant papers:

- Books, gray literature, surveys
- Non-peer reviewed (e.g. preprints)
- Non-English
- Duplicated studies

After filtering, 158 highly relevant studies were selected for in-depth review and analysis based on relevance to deep learning for malware detection. Each paper was critically read and analyzed to extract key information on algorithms, datasets, limitations, results etc. related to the use of deep neural networks for malware detection.

The structured literature screening process enabled methodical selection of the most pertinent prior studies on deep learning for malware detection across computing platforms.

2.3 Literature Screening Criteria

Searched 5 major digital libraries: IEEE Xplore, ACM DL, ScienceDirect, SpringerLink, Google Scholar.

Initial search results filtered using inclusion criteria:

- ✓ Peer-reviewed journal or conference papers
- ✓ Published between 2015-2023
- ✓ English language
- ✓ Full text available

Exclusion criteria to remove irrelevant papers:

- ✓ Books, gray literature, surveys
- ✓ Non-peer reviewed (e.g. preprints)
- ✓ Non-English
- ✓ Duplicated STUDIES

After filtering, 158 highly relevant studies were selected for in-depth review and analysis based on relevance to deep learning for malware detection. Each paper was critically read and analyzed to extract key information on algorithms, datasets, limitations, results etc. related to the use of deep neural networks for malware detection.

The structured literature screening process enabled methodical selection of the most pertinent prior studies on deep learning for malware detection across computing platforms.

2.4 Inclusion and Exclusion Criteria

The search across five literature databases initially returned 935 total papers. We refined the list in Table 2to identify the most relevant publications for our review. Papers were excluded based on screening of titles, abstracts, document types, languages, and if deemed irrelevant after full-text review. Specific exclusion criteria were survey papers, book chapters, gray literature, duplicates, non-peer reviewed publications, and non-English papers. By systematically applying these criteria, we filtered the initial 935 papers down to 158 highly relevant journal articles to review and analyze within our domain of deep learning for malware detection. The inclusion and exclusion criteria enabled us to hone in on the most pertinent literature from the initial search results.

Table 2: Inclusion and Exclusion Criteria

Inclusion Criteria
The goal of this research is a deep learning-based malware or intrusion detection system
The paper is either a preview or an article published in a peer-reviewed scientific journal.
Between January 2015 and December 2023, three issues of the magazine will be published.
Exclusion Criteria
First, there is information that examines the economic, commercial, and legal consequences of malware and intrusion detection.
Reports and blogs are two forms of gray literature.
Third, any documents that are not in English.
Critical evaluations.
Use of two-sided paper is number five
This category includes non-journal papers, such as those given at conferences.
There is a scarcity of work on malware detection that does not use deep learning.

3. LITERATURE REVIEW

3.1 Windows Malware Detection using Deep Learning Techniques

Ni et al.[26] presented a convolutional neural network-based "Malware Classification Using Sim-Hash and CNN" (MCSC). They decompile the infecting code and utilize the grayscale pictures that arise to identify malware families. To transform comparable viral code into hash values, locality-sensitive hashing (LSH) is utilized. The hash values are then transformed into grayscale pictures for neural network training. They claim that their technology detects malware at a rate of 98% or higher.

Zhao et al.[27] describe MalDeep as a deep learning-based malware detection system that analyzes at the malware's binary file. Convolutional neural networks are used to categorize the pathogen once the binary file is transformed to a grayscale picture. One of their system's most impressive features is its 99% detection rate for dangerous malware.

A deep learning algorithm for malware detection that makes use of subtle system calls was developed by Zhang et al.[28]. Cuckoo sandbox monitors the specified program in order to obtain system call information and use it to train neural networks. Their method detects malware with 95% accuracy using simply system calls.

Zhang et al.[29] created a convolutional neural network model for detecting malware that decompiles the software into its component pieces to get op-codes and API 133 calls. Each binary is organized, and the API frequency vectors and PCA-initialized opcode bigram matrices are constructed. These data are used to train a convolutional neural network (CNN) and a backpropagation neural network (BPNN) to include features. Their malware detection technology has a 95% accuracy rate.

Zhong and Gu [30]demonstrated a multi-tiered deep learning strategy for picking significant characteristics from static and dynamic feature sets. It generates cluster sub-trees by grouping comparable qualities together using the K-means algorithm. It decides if an application is dangerous or safe by merging the outputs of the deep learning models in the tree.

The ransomware detection approach presented by Zhang et al.[31] converts ransom ware family names and op-code information into numerical tensors in order to train a neural network. Their approach employs self-attentional convolutional neural networks (SA-CNN). One disadvantage is that the accuracy is just about 90%.

Deep learning has become a prominent technique to protect Windows systems against malware, with convolutional neural networks applied extensively. Accuracy rates up to 99% have been achieved by researchers in detecting new malware samples. But challenges like improving detection of ransomware illustrate that continued advancement of deep learning systems can further enhance malware detection on the Windows platform.

In [32], Yuxin and Siyi developed a deep belief network approach for malware detection that extracts opcode sequences from malware executable. A PE parser is used in their system to convert the PE file into a set of machine instructions. A feature extractor finds high-classification-power n-gram sequences and utilizes them to represent the PE file as an n-gram vector. This data is sent into a malware detection system that employs neural networks. Their approach detects dangerous malware with a 98% success rate.

Yue [33] suggests this loss function for malware photo identification using deep convolutional networks by combining softmax regression and entropy loss. They argue that their loss function appropriately handles the challenges raised by datasets with significantly variable malware family distributions.

It was first used by Ye et al. [34] as a malware detection technique that operates directly on Windows PE files. An API feature extractor is employed in their suggested approach to decompress the PE file and extract the relevant API calls. It uses constrained Boltzmann machine-based deep learning models and unsupervised heterogeneous auto-encoders to identify malware based on API request patterns.

Xiaofeng et al. [35] proposed an LSTM RNN malware detection approach that integrated machine learning and deep learning. It collects API call sequences and statistical statistics from sandboxed malware executables. Before the system call sequences are fed into the deep LSTM model for malware classification, they are first categorized using a random forest model.

ScalMalNet is a distributed system designed by Vinayakumar et al.[36] to gather malware samples from multiple websites. These samples are processed in an immediate or asynchronous distributed manner. They recommended detecting malware using image processing and static and dynamic analysis. According to their research, deep learning malware detection is considerably more successful than classic ML approaches.

A convolutional neural network technique was presented by Kolosnjaji et al.[37] for detecting malware in binary files. Grayscale graphics are created by breaking down the malware binary into 8-bit chunks, which are then converted to decimal values ranging from 0 to 255, organized into a 2D array, and displayed. With the assistance of this image, CNN learns to spot infections.

Athiwaratkun and Stokes[38]proposed MalConv, using 1D convolutions on raw byte sequences for malware detection without feature engineering. It views the malware binary as a long input sequence, applying narrow 1D convolutions and max-pooling to automatically learn local relationships between malware bytes.

Deep learning techniques like DBNs, CNNs, and LSTMs have been extensively explored for Windows malware detection using static and dynamic analysis of PE files, opcodes, and API call sequences. Direct modeling of malware binaries as images or sequences enables deep learning to achieve high accuracy without relying on manual feature extraction.

Convolutional neural networks were used by Anderson et al. [39] to develop a deep learning-based malware detection system. Their approach accepts raw byte sequences as input rather than relying on manual feature engineering. Malware binary files are converted into byte plots and visualized as grayscale images to train the convolutional networks. This spatial representation helps model positional relationships within the malware code.

Yousefi-Azar et al.[40]developed a self-taught learning system using sparse autoencoders for detecting malicious Windows executables. They generate image inputs from binary file hashes and apply transformations to augment the training data. This improves the model's ability to generalize to new malware samples.

Han et al.[41] created a malware detection framework using raw Windows dynamic trace log data as input to a deep neural network. By eliminating manual feature extraction and using native log data, their system achieves higher accuracy compared to classical machine learning techniques.

A long-term study on deep learning for just-in-time malware detection in Windows executables was presented by Rhode et al.[42]. They evaluate detection performance over an extended period as new malware samples appear. Deep learning consistently outperforms traditional machine learning approaches over time as new threats emerge.

These and other studies highlight the capabilities of deep learning to enable robust malware detection in Windows without extensive feature engineering. Deep neural networks can automatically learn complex positional relationships and sequences found in malware code. Their ability to generalize from raw binaries and log data also improves detection of new malware strains over time. Overall, deep learning shows significant promise for enhancing Windows malware detection.

3.2 Android Malware Detection using Deep Learning Techniques

Researchers used deep learning to construct malware and intrusion detection systems for the Android platform, similar to their work on Windows. Meta-data from the literature on Windows-based malware detection and data highlighted in the research questions (RQ7 and RQ8) are used to analyze the work conducted on the Android platform in this section.

Devi [65]offered a solution for identifying fraudulent applications that needed user consent on Android. They collected data from Android package manifests and permissions, created feature vectors, and trained their model using neural networks and the k-means clustering technique. The method's low success rate (88% to be exact) is a disadvantage.

Karbab et al.[66] presented MalDozer, a tool for detecting Android malware based on the sequence of API method calls.MalDozer extracts API method calls by using classes from an Android package. The dex file is subjected to a process of discretization, wherein an identification is assigned to each API method, resulting in the creation of semantic vectors. In this study, researchers utilize neural networks to predict the possible risks associated with Android applications. One advantage of their methodology is in its ability to consistently get a high F1 score across several datasets. Khedkar et al. [67] presented a methodology for detecting Android malware that relies on permissions. By employing network clustering techniques, the application organizes its attributes and generates a dataset for the purpose of categorizing newly occurring illnesses.

API methods, opcode features, authorization features, shared library function op-code features, component features, and environmental elements were all included in the feature vectors that Kim et al.[68] constructed for each feature. The malware categorization model is further enhanced by incorporating these vectors. By including several aspects instead of relying on a limited number, they possess a competitive edge over other methodologies.

A deep learning-based malware prediction system that takes CPU, memory, and battery usage into account was proposed by Milosevic and Huang[69]. Their unsupervised technique collects data using encoder-decoder and LSTM networks and runs on a variety of platforms. Their system's weakness is reflected in a low F1 score, which hovers around 80%.

Yuan et al.[70] extracted three elements to construct an online Android malware detection tool: crucial permissions, sensitive API calls, and dynamic behavior. These are inputs to deep belief networks, which aid in the detection of infections in apps. Following an unsupervised training period, their model is enhanced via supervised backpropagation.

Yuan et al.[71] developed an approach that makes advantage of API sensitivity, permissions, and dynamic behavior with their work on deep learning. Their method effectively categorizes malware with a success rate of over 96% by utilizing deep belief networks and a total of over 200 features.

Yen and Sun[72] provided a technique for identifying malware in APK files based on the significance of terms. To assign a value to each word in the APK's translated Java classes, text mining using Term Frequency-Inverse Document Frequency (TFIDF) is employed. A CNN architecture generates word-significant pictures.

Xie et al.[73] extracted seven types of malware characteristics using a CNN technique, including APIs, hardware, intents, permissions, and limited APIs. To train and validate CNNs, they first create feature vectors, then turn them into matrices and split the dataset. The 99.25% accuracy is pretty acceptable.

Wang et al.[74] combined a deep autoencoder with a customized CNN known as CNN-S to detect Android viruses. To train the model, they employ seven different sorts of characteristics, such as limited APIs, permissions, intents, and coding

patterns. Its 99.82% accuracy percentage is a big plus.

Luo et al. [75]introduced ITMF for analyzing and discovering Android malware using picture texture median filtering. The lower noise levels in ITMF improve both image and signal processing. The grayscale photos are supplied to ITMF for feature extraction once the binary images have been transformed to vectors. Deep belief networks outperform shallow learning when trained on properties such as APIs and URLs.

Saif et al. [76] created a deep belief network system by analyzing static and dynamic Android applications. Relief feature selection is used to reduce the number of features that contribute to a vector, which may include manifest nodes, API calls, system functions, and dynamic behavior. This vector is utilized in the construction of a deep learning network classifier. Using API call graphs, Pektas and Acarman[77] presented a technique for detecting Android malware. The model is fed graph embedding vectors after a given number of consecutive API requests. This simplifies the extraction of features from API call patterns for use in malware classification. Deep learning has been intensively researched for static, dynamic, and hybrid analysis of Android malware. The use of neural networks in conjunction with data mining and filtering procedures results in excellent accuracy without the requirement for human feature engineering.

Pektas and Acarman [78] created a method for detecting Android malware that uses features from instruction call graphs to look at every possible path of execution. Their method derives call trees and execution routes in terms of opcodes via pseudo-dynamic analysis. Graphs of potential paths are created, which are subsequently translated into numerical vectors. These vectors are given into a model that identifies malware risk using Long Short-Term Memory Recurrent Neural Networks. The results demonstrated that they were more accurate than typical machine learning approaches.

Nauman et al. [79] investigated numerous deep learning algorithms for Android malware detection at scale, including CNNs, DBNs, LSTMs, and autoencoders. Static analysis and manifest files give information like as components, limited APIs, and deep model rights. The efficiency of various architectural layouts was evaluated using malware feature data. A.

Martın et al.[80] presented CANDYMAN, which combines deep learning, dynamic analysis, and Markov chains to classify Android malware. With DroidBox, you may gather information on the network, files, courses, and SMS services in real time. These data are fitted with a Markov chain model to create feature vectors that are fed into deep neural networks for classification. Deep learning and machine learning experiments, on the other hand, demonstrated just a moderate gain in accuracy (approximately 81%).

Shiqi et al.[81] developed an attention-CNN-LSTM model to extract texture fingerprints and malware activity embeddings from binaries using deep belief networks. Malicious applications create grayscale pictures. The attention-CNN-LSTM architecture receives the fingerprint properties and activity embeddings required for malware identification. They outperformed typical machine learning algorithms in terms of precision.

The use of techniques such as dynamic analysis, call graph mapping, opcode extraction, permissions, and API analysis has resulted in improvements in Android malware detection using deep learning. Long short-term memory networks that combine CNNs with attention mechanisms and binary image converters are also promising. However, further research is needed to increase generalizability and accuracy in the face of expanding mobile threats.

An Android malware detection method using the Bag of Words paradigm to extract hardware, permissions, APIs, intents, and network address characteristics was proposed by Halim et al. [82]. A convolutional neural network (CNN) and a long short-term memory (LSTM) stack were examined as deep learning architectures. When tested for malware, both the CNN-LSTM and the LSTM-CNN obtained 98.53% accuracy.

A deep belief network strategy utilizing Lasso feature selection and shrinkage is presented by Elsersy and Anuar [83]. They compared a KNN classifier to a DBN classifier for malware detection and discovered that the latter was more accurate. We did, however, set a limit on total accuracy, which came in at 85.22%.

Using API call sequences, D'Angelo et al.[84] generated sparse matrices to simulate temporal behavior in an autoencoder system. Features collected from sparse matrices are used to distinguish between malicious and genuine software.

For the goal of recognizing Android malware, Chen et al.[85]recommended modeling permission and API features as word vectors using word2vec.

Amin et al. [86] combined DBNs, LSTMs, CNNs, and autoencoders with other deep learning algorithms to develop a method for identifying Android malware based on.dex files. They claim that their study of byte code attributes can properly classify malware with 99.9% certainty.

When Android apps are in operation, Alzaylaee et al. [87] present a system that employs DynaLog dynamic analysis to extract components such as APIs, actions, and permissions. The highest-scoring features from InfoGain are fed into deep learning models that detect malware.

3.3 IoT Malware detection using deep learning techniques

Many methods and strategies for recognizing malicious files and IoT malware are documented in the literature. Specific strategies combine static and dynamic analytic capabilities to detect Android-based mobile malware. Continuous research is being conducted to establish a categorization model to investigate the relationship between potential hazards and vulnerabilities in home automation systems[155]. The security of smartphones in the context of the Internet of Things, as well as the detection of application threats and impacts, are also being investigated.

In terms of detection methods, IoT malware detection approaches can be divided into two primary categories: dynamic and static analysis.

Future research will focus on analyzing IoT security difficulties, problems, and challenges and discussing security objectives, aims, and vulnerabilities. Various detection strategies based on virus characteristics, tracking of hazardous actions, and energy usage are being researched to reduce the threat of IoT malware[160]. Malware detection is also done automatically using machine learning methods such as ensemble classifiers and the ADA GRAD optimize algorithm. These strategies strive to recognize application attributes and classify them as risky, aggressive, benign, or malicious to safeguard IoT networks from malware assaults and improve the overall security and stability of the Internet[161]. Given the expanding number of Internet of Things devices and their critical importance in many applications, efficient malware detection and mitigation approaches are critical. The proposed architecture for developing and recognizing new IoT malware samples at the edge layer of IoT networks using raw byte code is a potential solution to the problem of a scarcity of malware samples for machine learning-based detection approaches [162].

However, there is insufficient literature specifically concentrated on IoT malware. Despite this scarcity, analysts have identified it as a substantial threat to internet security and stability. They stress the importance of comprehending IoT malware through analysis and detection to mitigate 2022 to 2023 effectively, and it is evident that there is a growing concern about the threat of IoT malware and a need for effective detection and mitigation techniques.

Machine learning, specifically deep learning techniques, is widely explored for IoT malware detection and classification. These techniques have shown great potential in improving the accuracy and efficiency of IoT malware detection compared to traditional methods. However, there is a lack of research on IoT malware analysis, and most existing studies use simple detection methods. Future studies should focus on developing advanced deep-learning models tailored explicitly for IoT malware detection.

Furthermore, the application of generative adversarial networks in developing deep learning models for identifying Android malware has been suggested. This also implies the possibility of investigating the use of generative adversarial networks for IoT malware detection. These advancements in detecting and analyzing malware demonstrate a growing recognition of IoT malware as a significant threat to Internet security. The review of existing literature emphasizes the importance of proactively identifying and mitigating IoT-based threats through advanced learning methods, accentuating the need for practical detection approaches that do not rely heavily on prior knowledge about malware features[163].

By conducting software analysis between IoT and Android samples and utilizing graph properties obtained from control flow graph structures, a detection system for IoT malware can be built using advanced deep learning models[164].

These techniques include utilizing deep learning algorithms and generative adversarial networks to detect and classify IoT malware and employing abstract graph structures, such as control flow graphs, for analyzing and detecting IoT malware. The use of generative adversarial networks to develop deep learning models for highly accurate identification of unknown malware samples has resulted in significant progress in IoT malware detection [165,166]. Furthermore, the system proposed for creating new malware samples at the edge layer of IoT networks utilizing raw byte code holds much promise. This could assist in alleviating the issue of malware sample scarcity for machine learning-based detection systems [167]. As IoT devices continue to increase and play a critical role in various applications, the emphasis on effective malware detection and mitigation methods becomes even more crucial[168].

3.4 Windows malware detection using the latest ML techniques

Malware targeting Windows operating systems is one of the most prevalent cybersecurity threats

today. Traditional signature-based antivirus solutions are inadequate in detecting new and evolving malware variants. However, machine learning techniques have emerged as a promising approach for detecting both known and novel malware[155].

Recent research has explored various machine learning algorithms for Windows malware detection, including deep learning neural networks, ensemble learning, and support vector machines. Deep learning methods like convolutional neural networks (CNNs) can automatically learn complex features from raw byte sequences of malware samples. Studies have shown that CNNs achieve over 99% detection accuracy on benchmark Windows malware datasets[151].

In addition to deep learning, ensemble methods like random forests and gradient boosted trees have also proven effective for Windows malware detection. Ensemble learners combine multiple weak predictive models to create an overall strong predictor. The random forest algorithm trains multiple decision trees on different subsets of features and data points, aggregating their outputs for the final classification. This provides robustness against overfitting on training data[152,153].

Research has also utilized support vector machines (SVMs) for malware classification. SVMs identify optimal hyperplanes to distinguish between malicious and benign software samples. Kernel functions like radial basis functions help SVMs classify complex malware types. SVMs achieve high accuracy, but their performance depends on careful feature engineering and selection[154,155]. So, modern machine learning has enhanced static, dynamic, and hybrid analysis of Windows malware. Static analysis focuses on characteristics extracted from the malware executable, while dynamic analysis executes the sample in a contained environment[156,157]. Hybrid analysis combines both for comprehensive detection[158]. Ultimately, an ensemble of multiple machine learning models provides optimal malware detection capabilities on the Windows platform[159,160].

3.5 Android malware detection using the latest ML techniques

Malware targeting the Android mobile operating system has exploded in recent years. Traditional malware scanners depend on malware signatures, and often fail to detect new threats that elude signature databases. However, machine learning presents a robust solution for identifying both known and zero-day Android malware.

Various machine learning algorithms have been leveraged for Android malware detection, including deep neural networks, logistic regression, naïve Bayes, random forests, and support vector machines. Deep learning models like deep belief networks (DBNs) and convolutional neural networks (CNNs) can automatically extract useful features from raw binaries and permissions[161,162].

Other techniques like logistic regression, naïve Bayes, and random forests rely on expert-defined features based on static and dynamic analysis. Key features includerequested permissions, intent filters, and function calls in the code. Dimensionality reduction methods like principal component analysis help eliminate redundant features. Support vector machines (SVMs) have proven particularly effective, as they can model complex Android malware families[163,164]. Most Android malware detection systems take a hybrid approach, combining static and dynamic analysis[165]. For instance, DBNs could first extract features from the Android application package (APK) code and manifest[166]. Then, an SVM or random forest algorithm could classify the app as malicious or benign based on those features[167,168].

So, machine learning has made Android malware detection scalable and automated[55]. Deep learning methods obviate manual feature engineering, while ensemble methods like random forest provide robust predictions. As Android malware continues to evolve, these AI-based techniques will grow increasingly important for security [59].

3.6 IoT Malware detection using the latest ML techniques

Internet of Things (IoT) devices are proliferating rapidly, but often lack adequate security. This makes them attractive targets for malware, including botnets like Mirai that compromise IoT devices for DDoS attacks. Machine learning presents a promising approach to detect malware infecting IoT devices like routers, IP cameras, and connected appliances[161].

A primary challenge with IoT malware detection is the diversity of hardware architectures and operating systems. ML techniques should be platform-agnostic to detect malware on Linux, RTOS, and other IoT OSes. Deep learning methods like convolutional neural networks (CNNs), recurrent neural networks (RNNs), and autoencoders can analyze raw binary files and network traffic on any platform[162].

In addition, IoT devices have limited computing capacity, precluding complex ML model deployment locally. Hence, ML-based IoT malware detection is best performed at the network level.

Network traffic analysis can identify anomalies and malicious connections. Supervised models like random forest and SVM can classify benign vs. malicious traffic when trained on labeled data samples[154,155].

Moreover, unsupervised ML algorithms like isolation forests, k-means clustering, and one-class SVM can detect IoT malware with no prior training data [60]. Such anomaly detection models learn patterns of normal behavior, flagging deviations as potential threats [63,64]. So, ML delivers efficient and robust IoT malware detection amidst hardware diversity and limited on-device capabilities [65,66]. Deep learning extracts useful features from binaries and network traffic, while ensemble methods classify threats. Anomaly detection techniques work even with limited labeled data[71]. As IoT adoption accelerates, ML will become indispensable to securing these devices against malware intrusions.

3.7 Linux Malware Detection

In addition to Windows, Android, and IoT devices, Linux-based systems are also vulnerable to malware threats[169]. As servers, desktops, and cloud infrastructure increasingly run on Linux, detecting Linux malware has become crucial. Signature-based antivirus tools are inadequate for detecting new Linux malware strains[170]. Machine learning provides robust Linux malware detection capabilities by modeling unique characteristics of Linux malware families[171].

Debnath S et al.[172] discussed the key features for Linux malware detection include executable metadata like format, headers, sections, libraries used etc. Dynamic analysis examines runtime behaviors like system calls, network activity, and file operations once executed in a contained environment. Hybrid approaches combine both static and dynamic features[172,173].

Shallow machine learning algorithms like logistic regression, naïve Bayes, support vector machines (SVMs), and random forests have been applied for Linux malware detection using expert-defined feature engineering[174]. Deep learning techniques like CNNs and RNNs can automatically extract useful features from raw binaries, disassembled code, assembly instructions and system calls[175].

Unsupervised learning is also relevant for Linux anomaly detection, as normal behavior can be profiled to detect deviations. One-class SVMs, isolation forests, and autoencoders identify anomalies without prior training[176]. Ensemble models that combine multiple shallow and deep learning algorithms also boost detection accuracy. To improve generalization across evolving Linux malware families, adversarial machine learning can augment training data with mutated malware samples. Transfer learning can leverage models trained on other platforms like Windows and Android and transfer knowledge to the Linux domain[177]. So, machine learning has emerged as a powerful tool for Linux malware detection amidst the rise of Linux adoption. Advanced deep learning and ensemble approaches overcome limitations of traditional signature-based methods. As malicious actors increasingly target Linux devices, robust ML-powered detection capabilities are crucial for security[178].

3.8 Main Deep Learning Algorithms in Malware Detection

We examined the application of various DL models in the literature[37] in order to address RQ4 and determine the primary deep learning algorithms utilized for malware detection. Table 2 summarizes the results, showing Convolutional Neural Networks (CNNs) were employed in over 50% of the surveyed papers[38]. This makes CNNs the most predominant technique for malware detection[39,40]. Various forms of LSTM-based neural networks were used in 25 studies, comprising 25% of the literature[41]. DBN and autoencoderbased algorithms were applied in 12% and 10% of publications, respectively[42,43].

Like many other domains, convolutional neural networks are the most popular deep learning approach for malware detection and classification[44]. CNNs can detect meaningful features from unsupervised data, making them well-suited for classification problems like image recognition, medical imaging, and malware detection[45]. In particular, CNNs are widely reported to be highly effective for image classification and object detection. The capability of CNNs to learn robust features from raw inputs like malware binaries and images enables their widespread use[46].

On the other hand, recurrent neural networks like LSTMs can model sequential data and time-based patterns in malware code and behavior. Variants of LSTM account for a significant portion of deep-learning malware research. Autoencoders and DBNs have shown promising results for learning compressed representations of benign and malicious files[47].The dominance of CNNs aligns with their demonstrated ability to learn spatial patterns from malware binaries represented as images or raw byte sequences. While other techniques have niche uses, convolutional neural networks are the workhorse of deep learning for robust malware detection across

most studies. The extensive use of CNNs highlights their applicability for learning features automatically from low-level malware data[151,152]. Sea Table 3.

RQ1 address windows PCs and Android mobile devices are the most common platforms targeted by malware, due to their widespread adoption. However, emerging research also looks at threats for Linux systems, IoT devices, and websites. On Windows, key malware threats include viruses, trojans, spyware, and ransomware. For Android, the open app ecosystem leads to risks from malicious apps containing backdoors, spyware, ransomware and banking trojans. Websites face threats like drive-by downloads and watering hole attacks that distribute malware. The prevalence of these platforms makes them prime targets for attackers. As their adoption continues growing, securing them from malware threats is crucial.

RQ2 shows the peak era of deep learning malware detection research is from 2015-2023. Most studies have focused on Windows and Android malware, with top publication venues being IEEE Transactions on Information Forensics and Security, Computers & Security journal, and IEEE Access journal. The surge in deep learning research for malware coincides with the emergence of AI/ML across security domains. CNNs and other deep learning methods allow learning from raw malware samples like binaries and bytecode without extensive feature engineering. Their ability to automate feature extraction and train on low-level malware data has driven adoption for detecting constantly evolving threats.

RQ3 address the most widely used deep learning techniques are convolutional neural networks (CNNs), recurrent networks like LSTM, deep belief networks (DBNs), and autoencoders. CNNs can directly learn spatial patterns and relationships from raw binaries and opcode sequences. LSTMs and GRUs model the sequential nature of malware behaviors over time. DBNs help learn hierarchical abstract features from malware. Auto-encoders allow learning compact latent representations of malware. These techniques enable end-toend learning from low-level malware executables, API calls, opcodes, etc. without relying on manual feature extraction and selection.

RQ4 shows that out of the deep learning algorithms, convolutional neural networks dominate, applied in over 50% 620 of papers. RNN/LSTM networks are also popular for sequential data. Autoencoders have niche uses for anomaly detection. DBNs help extract hierarchical features. The prevalence of CNNs highlights their ability to learn spatial relationships from malware binaries represented as images or sequential data. They can effectively model positional patterns in malware code and binaries where feature location matters.

RQ5 address the key challenges faced in applying deep learning for malware detection include limited labeled training data, model overfitting on seen malware families, evasion attacks degrading generalization, and lack of model interpretability. Sustaining accuracy over long periods as new malware strains continuously evolve is an open research problem. Adversarial malware can craft inputs to evade detection. Hybrid deep learning ensemble models help improve robustness. But a universal robust solution remains lacking. Lack of transparency around model logic also makes real-world deployment difficult.

RQ6 shows the most Android malware detection studies do not focus on sustainability, automatic selection of optimal algorithms, and continuous retraining over time. But adaptability over long periods is critical as the malware landscape rapidly evolves. Incremental learning to update models and retaining performance over years remains an open challenge. Automated selection of the best performing deep learning architectures is also lacking.

RQ7 shows the static, dynamic, and hybrid analysis techniques are used to extract features from Android apps for detecting new malware variants. However, more research is needed to strengthen zero-day threat detection. Generative adversarial networks show promise for improving generalization. But evasive malware continues to be a challenge. Signatures and heuristics have limited effectiveness for brand new threats. So, techniques to boost resilience are crucial.

RQ8 address the widely used datasets for evaluating Android malware detection include Drebin, Contagio, VirusShare and the Android Genome Project. For Windows malware, common datasets are EMBER, BIG2015 and SOREL-20M. Standard datasets allow comparing different techniques. But they may not reflect realworld diversity. Expanding datasets with adversarial samples can help improve robustness. Overall the lack of rich labeled real-world data remains a key limitation.

4. DISCUSSION

4.1 Effectiveness of Deep Learning in Malware Detection

Deep learning works best when it is used to analyze unstructured data. We need to either

organize the data or create systems that can evaluate unstructured data because the majority of the data produced by these systems is unstructured and comes in different forms. Deep learning can be used to develop malware detection algorithms that work better with unstructured and unlabeled data. Furthermore, if the raw data supplied accurately represents the problem, a deep learning system can quickly complete thousands of challenging and repetitive tasks after only one training session. The lack of machine learning or deep learning algorithms in traditional malware detection systems makes them ineffective in identifying new malware strains. "Malware definitions," which are used to identify potential threats, are frequently updated by them.

However, after the training, machine learning and deep learning algorithms can identify complex patterns in both structured and unstructured data, which is essential for creating malware detection systems that are effective. Malware programmers create programs that may undergo code modifications during transmission, rendering them undetected by typical pattern-matching tools. These viruses are smart and simple, readily tricking pattern-matching programs. Many malware samples have similar behavioral characteristics that ML and DL algorithms may use to uncover previously undiscovered malware.

4.2. Performance of DL Compared with ML

When trained on large datasets, deep learning algorithms outperform machine learning algorithms in terms of output accuracy. High-level attributes may be inferred using these techniques without the need for time-consuming feature extraction or specific subject knowledge. Deep learning was found to be more effective than machine learning in various papers that we re-examined, which used both machine learning and deep learning techniques to detect malware [101,102].

Early-stage malware detection during the first few seconds of a program's execution was developed by Rhode et al. [39]. They compared RNN to common learning algorithms like SVM (support vector motion) and found that RNN performed better. While SVM's 80% accuracy was rather good, RNN's 96% accuracy after 19 seconds was far superior. The accuracy rate of Decision Trees was 92.6%, whereas that of the Random Forest classifier was 92%. Haddad-pajouh et al. [103] employed RNN-LSTM to identify risks in an IoT setting with a success rate of 98.18%. They also used more traditional forms of machine learning, with KNN yielding the highest accuracy (94%) of the bunch.

An automotive cyber-attack intrusion detection system with an RNN accuracy of 86.9% was created by Loukas et al. [114]. They obtained accuracies of 73.3%, 74%, 77.3%, and 79.9%, respectively, using a range of machine learning techniques, including Logistic Regression, D.T., R.F., and SVM. (The Ullah et al. [102] cyber threat detection system was far more accurate 96% than earlier systems that relied on machine learning algorithms.

Vinayakumar et al. [109] developed an intrusion detection system that relied on deep neural networks and achieved an accuracy of 99.2%. Using conventional machine learning methods, we were able to achieve an average accuracy of almost 80%. Luo et al. [81] used a method called attention CNN-LSTM to detect Android malware. They observed that the average accuracy of their deep learning-based model was 96%, whereas that of the SVMand KNN-based models was 95% and 94%, respectively.

Pektas and Acarman [78] proposed a comparable methodology for identifying Android malware, which involves the utilization of instruction call graphs. Their study yielded a 91.4% accuracy rate. The suggested approach demonstrated higher accuracy compared to many commercially available learning algorithms (KNN, Logistic Regression, SVN, and R.F.), as evidenced by percentages of 80%, 70%, 79%, and 89%.

Schranko de Oliveira and Sassi [90] have developed an Android malware detection system using a deep neural network. This system outperforms other machine learning Regression, Extra Trees, and K-Nearest Neighbors (KNN), with an accuracy rate of 91%. The study conducted by Jain et al. [55] showed that the accuracy achieved using Extreme Learning Machines (ELM) with a single hidden layer was 97.7%, which surpassed the accuracy obtained by a Convolutional Neural Network (CNN) architecture, which was 96.3%.

Similarly, Pastor et al. [118] compared many classical learning algorithms to CNN and found that conventional learning algorithms generated equivalent or higher results when recognizing crypto-mining activities. In most cases, deep learning models significantly outperformed their non-deep counterparts [103,104]. These numbers point to the efficacy of deep learning systems in detecting and pursuing threats like malware. We may not find a highly precise scalable solution using only shallow learning methods [105]. Nonetheless, as previous studies and the results of our experiment demonstrate, DL algorithms are not guaranteed to outperform ML techniques [106]. We compared the accuracy of Deep Autoencoders to that

of many other ML techniques and found that machine learning models performed better[107,108].

4.3. Challenges in Malware Detection Using Deep Learning

To train machine learning and deep learning algorithms, a large amount of data is required [109]. One of the most difficult difficulties in malware and threat detection is providing the algorithm with enough harmful and benign samples[110]. It is critical to keep the public datasets[111] up to date so that models may be trained using the most recent malware samples.

There is also the issue of "overtraining the model," which might result in incorrect findings. This might happen if the data is noisy or if inaccurate labeling is possible [112]. Several studies have demonstrated high levels of accuracy[113,114]. They did not, however, present any experimental evidence of their systems' resiliency to new malware threats [115]. To benefit from the advantages of deep learning over standard threat detection systems [125,126], malware detection systems must be able to differentiate between unique malware types and variants on the malware samples used for training.

The fast expansion of the Android ecosystem and the associated multiplication of issues [139,140] emphasize the need of a flexible approach that may be utilized regularly in the future to identify new types of dangers. To solve the issue of continually changing malware, the model only has to be altered every few years, at the expense of minor performance advantages. The model's evolvability is what makes it sustainable.

There are several deep learning algorithms in the literature that can deal with complicated challenges like malware classification across large data sets. However, most researchers fail to identify the best approach for their issues because they do not often train, test, and deploy the model related to ML algorithm (s) selection challenges [141]. Due to Android malware's dynamic nature and quick growth, maintaining up-to-date supervised detection models is a difficult task [142,144]. A long-term malware detection model must be created that can automatically update itself over time in an efficient and scalable manner. When determining a model's long-term viability, examine the retention rate, lifespan, and performance reduction after the specified time frame.

One of the key problems in using deep learning for autonomous feature engineering is picking or automatically learning features that will perform well over time and in the future[127]. Static, dynamic, and hybrid analytic techniques have been employed in the literature to automatically extract features for training the DL model[128]. The data quality is critical for machine learning and deep learning algorithms used to detect malware[129,130]. As a result, in addition to technological methodologies, the availability of a large and insightful dataset is important to the predicted accuracy of such systems [131,132].

5. CONCLUSIONS

This systematic literature review set out to provide a comprehensive overview of deep learning techniques applied for malware detection across computing platforms. Our analysis of 107 highly relevant studies from 2015-2023 reveals that convolutional neural networks dominate this research landscape, enabling effective learning of spatial patterns from raw malware samples. However, several limitations remain that constrain real-world deployment of deep learning-based malware defense systems.

A key objective was assessing the effectiveness of deep learning compared to traditional machine learning approaches relying on manual feature engineering. The literature overwhelmingly demonstrates enhanced accuracy from deep learning models like CNNs, LSTMs, and autoencoders that automatically extract useful features from executable files, byte sequences, and API calls. However, model resilience against obfuscation attacks that degrade generalization requires further improvement.

We also sought to identify key challenges faced in applying deep learning for robust malware detection. Insufficient labeled real-world training data, lack of model interpretability, and inability to sustain accuracy over long periods emerged as primary limitations. Though deep learning achieves high accuracy on benchmark datasets, performance in operational environments remains uncertain.

In conclusion, while deep learning shows significant promise for malware detection, progress on robustness, transparency, and continuous retraining is needed. As malware threats persist and evolve, developing sustainable, self-adaptive deep learning models must be a priority. This systematic review highlighted crucial gaps that need addressing to realize the potential of AI-powered techniques for reliable malware defense. More work is required to transition promising research solutions to large-scale real-world deployment.

6. FUTURE EXTENSIONS

Our research lead us to the following implications for readers and future researchers interested in malware detection using machine learning and deep learning:

- ✓ Because of the explosion of internet data, traditional data processing techniques cannot deal with the ensuing massive data quantities [133,134]. Big data frameworks like Hadoop and Spark enable the processing of enormous datasets [135,136]. Because huge volumes of data must be processed, internet security systems may benefit from merging deep learning virus detection with big data technologies [137].
- ✓ It remains uncertain how well deep learning malware systems proposed in research will perform at scale on big datasets. Testing and validation of big data is an open challenge [138].
- ✓ Web and internet security has not received as much attention in deep learning security research as the Windows and Android platforms have. However, internet security is just as important [139].
- ✓ Our study suggests developing deep learning systems for internet security should be an area of focus [140].
- ✓ Many studies report high malware detection accuracy, up to 99.9%, with deep learning. However, realizing this performance in real-world deployments remains an open problem. Researchers should enable easy and effective use of deep learning for malware protection by end users.
- ✓ Developing sustainable, self-evolvable deep learning models that avoid frequent retraining is important.

7. ACKNOWLEDGMENT

This work was supported by the Deanship of Scientific Research, Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia [Grant No. 5440].

REFERENCES:

[1]. N. Idika and A. P. Mathur, "A survey of malware detection techniques," Purdue University, vol. 48, no. 2, 2007

[2]. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, 2015.

[3]. M. Christodorescu, S. Jha, S. A. Seshia, D. Song, and R. E. Bryant, "Semantics-aware malware detection," in Proceedings of the 2005 IEEE Symposium on Security and Privacy (S&P'05), pp. 32–46, IEEE, Oakland, CA, USA, 2005 May

[4]. E. Gandotra, D. Bansal, and S. Sofat, "Malware analysis and classification: a survey," Journal of Information Security, vol. 05, no. 02, pp. 56–64, 2014.

[5]. A. S. Bist, "A survey of deep learning algorithms for malware detection," International Journal of Computer Science and Information Security, vol. 16, no. 3, 2018.

[6]. A. Naway and Y. Li, "A review on the use of deep learning in android malware detection," 2018, https://arxiv.org/abs/ 1812.10360.

[7]. T. Sonali Kothari and V. Khedkar, "Analysis of recent trends in malware attacks on android phone: a survey using scopus database," Library Philosophy and Practice, pp. 1–20, 2021.

[8]. B. Yadav and S. Tokekar, "Recent innovations and comparison of deep learning techniques in malware classification: a review," International Journal of Information Security Science, vol. 9, no. 4, pp. 230–247, 2021.

[9]. M. V. R. Kumar, S. Anand Kumar, A. Bando, S. R. Gs, H. Shah, and S. C. Reddy, "A survey of deep learning techniques for malware analysis," International Journal of Advanced Science and Technology, vol. 29, no. 4, pp. 6031– 6042, 2020.

[10]. H. Lubuva, Q. Huang, and G. C. Msonde, "A review of static malware detection for Android apps permission based on deep learning," International Journal of Computer Networks and Applications, vol. 6, no. 5, pp. 80–91, 2019.

[11]. D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim, "A survey of deep learning-based network anomaly detection," Cluster Computing, vol. 22, no. S1, pp. 949–961, 2019.

[12]. R. A. Ariyaluran Habeeb, F. Nasaruddin, A. Gani, I. A. Targio Hashem, E. Ahmed, and M. Imran, "Real-time big data processing for anomaly detection: a survey," International Journal of Information Management, vol. 45, pp. 289–307, 2019.

[13]. A. Souri and R. Hosseini, "A state-of-the-art survey of malware detection approaches using data mining techniques," Humancentric Computing and Information Sciences, vol. 8, no. 1, pp. 3–22, 2018.

[14]. S. Sharma and A. Kaul, "A survey on Intrusion Detection Systems and Honeypot based proactive security mechanisms in VANETs and VANET Cloud," Vehicular communications, vol. 12, pp. 138–164, 2018.

[15]. J. Mart´ınez Torres, C. Iglesias Comesaña, and P. J. GarciaNieto, "Review: machine learning techniques applied to cybersecurity," International Journal of Machine Learning and Cybernetics, vol. 10, no. 10, pp. 2823– 2836, 2019.

[16]. S. Hajiheidari, K. Wakil, M. Badri, and N. J. Navimipour, "Intrusion detection systems in the Internet of things: a Security and Communication Networks comprehensive investigation," Computer Networks, vol. 160, pp. 165–191, 2019.

[17]. R. Coulter and L. Pan, "Intelligent agents defending for an IoT world: a review," Computers Security, vol. 73, pp. 439–458, 2018.

[18]. E. Benavides, W. Fuertes, S. Sanchez, and M. Sanchez, "Classification of phishing attack solutions by employing deep learning techniques: a systematic literature review," Developments and advances in defense and security, pp. 51– 64, 2020.

[19]. K. Bakour, H. M. Unver, and R. Ghanem, "(e Android ¨ malware detection systems between hope and reality," SN Applied Sciences, vol. 1, no. 9, pp. 1120–1142, 2019.

[20]. M. Aly, F. Khomh, M. Haoues, A. Quintero, and S. Yacout, "Enforcing security in Internet of (ings frameworks: a systematic literature review," Internet of =ings, vol. 6, Article ID 100050, 2019.

[21]. A. M. Aleesa, B. B. Zaidan, A. A. Zaidan, and N. M. Sahar, "Review of intrusion detection systems based on deep learning techniques: coherent taxonomy, challenges, motivations, recommendations, substantial analysis and future directions," Neural Computing & Applications, vol. 32, no. 14, pp. 9827–9858, 2020.

[22]. Z. Wang, Q. Liu, and Y. Chi, "Review of android malware detection based on deep learning," IEEE Access, vol. 8, pp. 181102–181126, 2020.

[23]. B. Kitchenham, "Procedures for Performing Systematic Reviews," Keele UK Keele University, vol. 33, pp. 1– 26, 2004.

[24]. B. Kitchenham and S. Charters, "Guidelines for Performing Systematic Literature Reviews in Software Engineering,"2007.

[25]. S. Keele, "Guidelines for performing systematic literature reviews in software engineering," vol. 5, EBSE, Mumbai,India, 2007, Technical report Ver. 2.3 EBSE Technical Report.

[26]. S. Ni, Q. Qian, and R. Zhang, "Malware identification using visualization images and deep learning," Computers & Security, vol. 77, pp. 871–885, 2018.

[27]. Y. Zhao, C. Xu, B. Bo, and Y. Feng, "Maldeep: a deep learning classification framework against malware variants based on texture visualization," Security and Communication Networks, vol. 2019, Article ID 4895984, 11 pages, 2019.

[28]. J. Zhang, K. Zhang, Z. Qin, H. Yin, and Q. Wu, "Sensitive system calls based packed malware variants detection using principal component initialized MultiLayers neural networks," Cybersecurity, vol. 1, no. 1, pp. 10–13, 2018.

[29]. J. Zhang, Z. Qin, H. Yin, L. Ou, and K. Zhang, "A featurehybrid malware variants detection using CNN based opcode embedding and BPNN based API embedding," Computers & Security, vol. 84, pp. 376–392, 2019.

[30]. W. Zhong and F. Gu, "A multi-level deep learning system for malware detection," Expert Systems with Applications, vol. 133, pp.162, 2019.

[31]. B. Zhang, W. Xiao, Xi Xiao, A. K. Sangaiah, W. Zhang, and J. Zhang, "Ransomware classification using patch based CNN and self-attention network on embedded N-grams of opcodes," Future Generation Computer Systems, vol. 110, pp. 708–720, 2020.

[32]. D. Yuxin and Z. Siyi, "Malware detection based on deep learning algorithm," Neural Computing and Applications, vol. 31, pp. 461–472, 2017.

[33]. S. Yue, "Imbalanced malware images classification: a cnn based approach," 2017,

[34]. Y. Ye, L. Chen, S. Hou, W. Hardy, and X. Li, "DeepAM: a heterogeneous deep learning framework for intelligent malware detection," Knowledge and Information Systems, vol. 54, no. 2, pp. 265–285, 2018.

[35]. L. Xiaofeng, Z. Xiao, J. Fangshuo, Y. Shengwei, and S. Jing, "ASSCA: API based sequence and statistics features combined malware detection architecture," Procedia Computer Science, vol. 129, pp. 248–256, 2018.

[36]. R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, and S. Venkatraman, "Robust intelligent malware detection using deep learning," IEEE Access, vol. 7, pp. 46717–46738, 2019.

[37]. S. Venkatraman, M. Alazab, and R. Vinayakumar, "A hybrid deep learning image-based analysis for effective malware detection," Journal of Information Security and Applications, vol. 47, pp. 377–389, 2019.

[38]. M. Tang and Q. Qian, "Dynamic API call sequence visualisation for malware classification," IET Information security,vol. 13, no. 4, pp. 367–377, 2019.

[39]. M. Rhode, P. Burnap, and K. Jones, "Early-stage malware prediction using recurrent neural networks," Computers & Security, vol. 77, pp. 578–594, 2018.

[40]. M. F. Rafique, M. Ali, A. S. Qureshi, A. Khan, and M. Mirza, "Malware classification using deep learning based feature extraction and wrapper based feature selection technique," 2019,

[41]. M. H. Nguyen, D. L. Nguyen, X. M. Nguyen, and T. T. Quan, "Auto-detection of sophisticated malware using lazy-binding control flow graph and deep learning," Computers & Security, vol. 76, pp. 128–155, 2018.

[42]. A. Namavar Jahromi, S. Hashemi, A. Dehghantanha et al., "An improved two-hidden-layer extreme learning machine for malware hunting," Computers & Security, vol. 89, Article ID 101655, 2020.

[43]. Q. Le, O. Boydell, B. Mac Namee, and M. Scanlon, "Deep learning at the shallow end: malware classification for nondomain experts," Digital Investigation, vol. 26, pp. S118–S126, 2018.

[44]. J. Y. Kim, S. J. Bu, and S. B. Cho, "Zero-day malware detection using transferred generative adversarial networks based on deep autoencoders," Information Sciences, vol. 460-461, pp. 83–102, 2018.

[45]. M. Kalash, M. Rochan, N. Mohammed, N. Bruce, Y. Wang, and F. Iqbal, "A deep learning framework for malware classification," International Journal of Digital Crime and Forensics, vol. 12, no. 1, pp. 90–108, 2020.

[46]. S. Huda, S. Miah, J. Yearwood, S. Alyahya, H. Al-Dossari, and R. Doss, "A malicious threat detection model for cloud assisted internet of things (CoT) based industrial control system (ICS) networks using deep belief network," Journal of Parallel and Distributed Computing, vol. 120, pp. 23–31,2018.

[47]. D. Gibert, C. Mateu, J. Planes, and R. Vicens, "Using convolutional neural networks for classification of malware represented as images," Journal of Computer Virology and Hacking Techniques, vol. 15, no. 1, pp. 15–28, 2019.

[48]. Z. Cui, F. Xue, X. Cai, Y. Cao, G. G. Wang, and J. Chen,"Detection of malicious code variants based on deep learning," IEEE Transactions on Industrial Informatics, vol. 14, no. 7, pp. 3187–3196, 2018.

[49]. Z. Cui, L. Du, P. Wang, X. Cai, and W. Zhang, "Malicious code detection based on CNNs and multi-objective algorithm," Journal of Parallel and Distributed Computing, vol. 129, pp. 50–58, 2019.

[50]. L. Chen, "Deep transfer learning for static malware classification," 2018, https://arxiv.org/pdf/1812.07606. Security and Communication Networks 27.

[51]. E. D. O. Andrade, J. Viterbo, C. N. Vasconcelos, J. Guerin, ´ and F. C. Bernardini, "A model based on lstm neural networks to identify five different types of malware," Procedia Computer Science, vol. 159, pp. 182–191, 2019.

[52]. A. F. Agarap, "Towards building an intelligent anti-malware system: a deep learning approach using support vector machine (SVM) for malware classification," 2017, https:// arxiv.org/abs/1801.00318.

[53]. E. Masabo, K. S. Kaawaase, and J. Sansa-Otim, "Big data: deep learning for detecting malware," in Proceedings of the 2018 IEEE/ACM Symposium on Software Engineering in Africa (SEiA), pp. 20–26, IEEE, Gothenburg, Sweden, 2018 May.

[54]. B. Yuan, J. Wang, D. Liu, W. Guo, P. Wu, and X. Bao, "Bytelevel malware classification based on Markov images and deep learning," Computers & Security, vol. 92, Article ID 101740, 2020.

[55]. Jain M, Andreopoulos W, Stamp M. CNN vs ELM for image-based malware classification. arXiv preprint arXiv:2103.13820. 2021 Mar 24.

[56]. F. O. Catak, A. F. Yazı, O. Elezaj, and J. Ahmed, "Deep learning based Sequential model for malware analysis using Windows exe API Calls," PeerJ Computer Science, vol. 6, Article ID e285, 2020.

[57]. Y. Fang, Y. Zeng, B. Li, L. Liu, and L. Zhang, "DeepDetectNet vs RLAttackNet: an adversarial method to improve deep learning-based static malware detection model," PLoS One, vol. 15, no. 4, Article ID e0231626, 2020.

[58]. H. Darabian, S. Homayounoot, A. Dehghantanha et al., "Detecting cryptomining malware: a deep learning approach for static and dynamic analysis," Journal of Grid Computing, vol. 18, no. 2, pp. 293–303, 2020.

[59]. R. Mitsuhashi and T. Shinagawa, "High-accuracy malware classification with a malware-optimized deep learning model," 2020.

[60]. D. Gibert, C. Mateu, and J. Planes, "HYDRA: a multimodal deep learning framework for malware classification," Computers & Security, vol. 95, Article ID 101873, 2020.

[61]. D. Vasan, M. Alazab, S. Wassan, B. Safaei, and Q. Zheng, "Image-Based malware classification using ensemble of CNN architectures (IMCEC)," Computers & Security, vol. 92, Article ID 101748, 2020.

[62]. M. Cho, J. S. Kim, J. Shin, and I. Shin, "Mal2d: 2d based deep learning model for malware detection using black and white binary image," IEICE - Transactions on Info and Systems, vol. 103, no. 4, pp. 896–900, 2020.

[63]. Y. Sung, S. Jang, Y.-S. Jeong, and J. H. J. J. Park, "Malware classification algorithm using advanced Word2vecbased BiLSTM for ground control stations," Computer Communications, vol. 153, pp. 342–348, 2020.

[64]. X. Huang, L. Ma, W. Yang, and Y. Zhong, "A method for windows malware detection based on deep learning," Journal of Signal Processing Systems, vol. 93, no. 2-3, pp. 265–273, 2021.

[65]. K. Devi, "Android malware detection using deep learning," International Research Journal of Engineering and Technology (IRJET), Academia, vol. 06, no. 05, 2019.

[66]. E. B. Karbab, M. Debbabi, A. Derhab, and D. Mouheb, "Android malware detection using deep learning on api method sequences," 2017,https://arxiv.org/abs/1712.08996.

[67]. Y. Suleiman, S. Sezer, and I. Muttik, "Android malware detection: An eigenspace analysis approach," in Pro- ceedings of the 2015 Science and Information Conference (SAI), pp. 1236–1242, 2015.

[68]. T. Kim, B. Kang, M. Rho, S. Sezer, and E. G. Im, "A multimodal deep learning method for android malware 836 detection using various features," IEEE Transactions on Information Forensics and Security, vol. 14, no. 3, pp. 773–788, 2019.

[69]. N. Milosevic and J. Huang, "Deep learning guided Android malware and anomaly detection," 2019, 839https://arxiv.org/ abs/1910.10660.

[70]. Z. Yuan, Y. Lu, and Y. Xue, "Droiddetector: android malware characterization and detection using deep learn-ing," Tsinghua Science and Technology, vol. 21, no. 1, pp. 114–123, 2016.

[71]. Z. Yuan, Y. Lu, Z. Wang, and Y. Xue, "Droid-sec: deep learning in android malware detection," in Proceedings of the 2014 ACM conference on SIGCOMM, pp. 371-372, Chicago, IL, USA, 2014 August.

[72]. Y. S. Yen and H. M. Sun, "An Android mutation malware detection based on deep learning using visualization of importance from codes," Microelectronics Reliability, vol. 93, pp. 109–114, 2019.

[73]. N. Xie, X. Di, X. Wang, and J. Zhao, "AndroMD: android malware detection based on convolutional neural networks," International Journal of Performability Engineering, vol. 14, no. 3, p. 547, 2018

[74]. W. Wang, M. Zhao, and J. Wang, "Effective android malware detection with a hybrid model based on deep autoencoder and convolutional neural network," Journal of Ambient Intelligence and Humanized Computing, vol. 10, no. 8, pp. 3035–3043, 2019.

[75]. S. Q. Luo, B. Ni, P. Jiang, S. W. Tian, L. Yu, and R. J. Wang, "Deep learning in Drebin: android malware image texture median filter analysis and detection," KSII Transactions on Internet and Information Systems (TIIS), vol. 13, no. 7, pp. 3654–3670, 2019.

[76]. D. Saif, S. Elgokhy, and E. A. Sallam, "Deep Belief Networksbased framework for malware detection in An- droid systems," Alexandria Engineering Journal, vol. 57, no. 4, pp. 4049–4057, 2018.

[77]. A. Pektas¸ and T. Acarman, "Deep learning for effective Android malware detection using API call graph em- beddings," Soft Computing, vol. 24, no. 2, pp. 1027–1043, 2020.

[78]. Pekta¸s A, Acarman T. Learning to detect Android malware via opcode sequences. Neurocomputing. 2020 Jul 5;396:599-608.

[79]. M. Nauman, T. Ali, S. Khan, and T. A. Syed, "Deep Neural Architectures for Large Scale Android Malware Analysis," Cluster Computing, vol. 21, pp. 1–20, 2018.

[80]. A. Martın, V. Rodr´ıguez-Fern´andez, and D. Camacho, "CANDYMAN: classifying Android malware families by modelling dynamic traces with Markov chains," Engineering Applications of Artificial Intelligence, vol. 74, pp. 121–133, 2018.

[81]. L. Shiqi, Z. Liu, B. Ni, H. Wang, H. Sun, and Y. Yuan, "Android malware analysis and detection based on attention-CNN-LSTM," Journal of Computers, vol. 14, no. 1, pp. 31–43, 2019.

[82]. M. A. Halim, A. Abdullah, and K. A. Z. Ariffin, "Recurrent neural network for malware detection," Int. J. Ad- vance Softwarw Computing. Appl, vol. 11, no. 1, pp. 43–63, 2019.

[83]. W. F. Elsersy and N. B. Anuar, "Android malware detection using deep belief network," PERTANIKA JOUR- NAL OF SCIENCE AND TECHNOLOGY, vol. 25, pp. 143–150, 2017.

[84]. G. D'Angelo, M. Ficco, and F. Palmieri, "Malware detection in mobile environments based on Autoencoders and API28 Security and Communication Networks images," Journal of Parallel and Distributed Computing, vol. 137, pp. 26–33, 2020.

[85]. T. Chen, Q. Mao, M. Lv, H. Cheng, and Y. Li, "Droidvecdeep: android malware detection based on Word2Vec and deep belief network," KSII Transactions on Internet and Information Systems (TIIS), vol. 13, no. 4, pp. 2180–2197, 2019.

[86]. M. Amin, T. A. Tanveer, M. Tehseen, M. Khan, F. A. Khan, and S. Anwar, "Static malware detection and attrib- ution in android byte-code through an end-to-end deep system," Future Generation Computer Systems, vol. 102, pp. 112–126, 2020.

[87]. M. K. Alzaylaee, S. Y. Yerima, and S. Sezer, "DL-Droid: deep learning based android malware detection using real devices," Computers & Security, vol. 89, Article ID 101663, 2020.

[88]. X. Pei, L. Yu, and S. Tian, "AMalNet: a deep learning framework based on graph convolutional networks for malware detection," Computers & Security, vol. 93, Article ID 101792, 2020.

[89]. T. Lu, Y. Du, L. Ouyang, Q. Chen, and X. Wang, "Android malware detection based on a hybrid deep learning model," Security and Communication Networks, vol. 2020, Article ID 8863617, pp. 1–11, 2020.

[90]. A. Schranko de Oliveira and R. J. Sassi, "Chimera: an android malware detection method based on multimodal deep learning and hybrid analysis," 2020.

[91]. F. Mercaldo and A. Santone, "Deep learning for image-based mobile malware detection," Journal of Computer Virology and Hacking Techniques, vol. 16, no. 2, pp. 157–171, 2020.

[92]. M. Amin, D. Shehwar, A. Ullah, T. Guarda, T. A. Tanveer, and S. Anwar, "A deep learning system for health care IoT and smartphone malware detection," Neural Computing Applications, vol. 34, no. 14, pp. 11283– 11294, 2020.

[93]. X. Su, W. Shi, X. Qu, Y. Zheng, and X. Liu, "DroidDeep: using Deep Belief Network to characterize and detect android malware," Soft Computing, vol. 24, no. 8, pp. 6017–6030, 2020.

[94]. Z. Ma, H. Ge, Z. Wang, Y. Liu, and X. Liu, "Droidetec: android malware detection and malicious code localization through deep learning," 2020.

[95]. Z. Ren, H. Wu, Q. Ning, I. Hussain, and B. Chen, "End-toend malware detection for android IoT devices using deep learning," Ad Hoc Networks, vol. 101, Article ID 102098, 2020.

[96]. W. Niu, R. Cao, X. Zhang, K. Ding, K. Zhang, and T. Li, "OpCode-level function call graph based android malware classification using deep learning," Sensors, vol. 20, no. 13, p. 3645, 2020.

[97]. R. Feng, S. Chen, X. Xie, G. Meng, S. W. Lin, and Y. Liu, "A performance-sensitive malware detection system using deep learning on mobile devices," IEEE Transactions on Information Forensics and Security, vol. 16, pp. 1563–1578, 2021.

[98]. H. Zhu, Y. Li, R. Li, J. Li, Z. H. You, and H. Song, "SEDMDroid: an enhanced stacking ensemble framework for android malware detection," IEEE Transactions on Network Science and Engineering, vol. 8, no. 2, pp. 984–994, 2021.

[99]. J. Feng, L. Shen, Z. Chen, Y. Wang, and H. Li, "A two-layer deep learning method for android malware detection using network traffic," IEEE Access, vol. 8, pp. 125786–125796, 2020.

[100]. Azmoodeh, A. Dehghantanha, and K. K. R. Choo, "Robust malware detection for internet of

(battlefield) things devices using deep eigenspace learning," IEEE transactions on sustainable computing, vol. 4, no. 1, pp. 88–95, 2019.

[101]. F. Xiao, Z. Lin, Y. Sun, and Y. Ma, "Malware detection based on deep learning of behavior graphs," Mathematical Problems in Engineering, vol. 2019, Article ID 8195395, pp. 1–10, 2019.

[102]. F. Ullah, H. Naeem, S. Jabbar et al., "Cyber security threats detection in internet of things using deep learning approach," IEEE Access, vol. 7, pp. 124379–124389, 2019.

[103]. H. Haddadpajouh, A. Dehghantanha, R. Khayami, and K. K. R. Choo, "A deep recurrent neural network based approach for internet of things malware threat hunting," Future Generation Computer Systems, vol. 85, pp. 88–96, 2018.

[104]. M. N. Al-Hawawreh, N. Moustafa, and E. Sitnikova, "Identification of malicious activities in industrial internet of things based on deep learning models," Journal of Information Security and Applications, vol. 41, pp. 1–11, 2018.

[105]. A. Abusnaina, M. Abuhamad, H. Alasmary et al., "A deep learning-based fine-grained hierarchical learning approachfor robust malware classification," 2020.

[106]. H. Naeem, F. Ullah, M. R. Naeem et al., "Malware detection in industrial internet of things based on hybrid image visualization and deep learning model," Ad Hoc Networks, vol. 105, Article ID 102154, 2020.

[107]. S. Lu, L. Ying, W. Lin et al., "New era of deeplearning-based malware intrusion detection: the malware detection and prediction based on deep learning," 2019, https://arxiv.org/abs/1907.08356.

[108]. B. Yu, J. Pan, D. Gray et al., "Weakly supervised deep learning for the detection of domain generation algorithms," IEEE Access, vol. 7, pp. 51542–51556, 2019.

[109]. R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," IEEE Access, vol. 7, pp. 41525–41550, 2019.

[110]. H. M. Song, J. Woo, and H. K. Kim, "In-vehicle networkintrusion detection using deep convolutional neural network," Vehicular Communications, vol. 21, Article ID 100198, 2020.

[111]. R. Priyadarshini and R. K. Barik, "A deep Learning Based Intelligent Framework to Mitigate DDoS Attack in Fog Environment," Journal of King Saud University-Computer and Information Sciences, vol. 34, no. 3, pp. 825–831, 2019.

[112]. A. Pektaş, and T. Acarman, "Deep learning to detect botnet via network flow summaries," Neural Computing & Applications, vol. 31, no. 11, pp. 8021–8033, 2019.

[113]. Y. Pan, F. Sun, Z. Teng et al., "Detecting web attacks with end-to-end deep learning," Journal of Internet Services and Applications, vol. 10, no. 1, pp. 16–22, 2019.

[114]. G. Loukas, T. Vuong, R. Heartfield, G. Sakellari, Y. Yoon, and D Gan, "Cloud-based cyber-physical intrusion detection formvehicles using deep learning," IEEE Access, vol. 6, pp. 3491–3508, 2018.

[115]. Y. S. Jeong, J. Woo, and A. R. Kang, "Malware detection on byte streams of pdf files using convolutional neural networks," Security and Communication Networks, vol. 2019,pp. 1–9, 2019.

[116]. Homayoun, A. Dehghantanha, M. Ahmadzadeh et al., "DRTHIS: deep ransomware threat hunting and intelligence system at the fog layer," Future Generation Computer Systems, vol. 90, pp. 94–104, 2019.

[117]. A. McDole, M. Abdelsalam, M. Gupta, and S. Mittal, "Analyzing cnn based behavioural malware detection techniques Security and Communication Networks 29 on cloud iaas," in Proceedings of the International Conference on Cloud Computing, pp. 64–79, Honolulu, HI, USA, 2020, September.

[118]. A. Pastor, A. Mozo, S. Vakaruk et al., "Detection of encrypted cryptomining malware connections with machine and deep learning," IEEE Access, vol. 8, pp. 158036–158055, 2020.

[119]. A. N. Jahromi, S. Hashemi, A. Dehghantanha, R. M. Parizi, and K. K. R. Choo, "An enhanced stacked LSTM method with no random initialization for malware threat hunting in safety and time-critical systems," IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 4, no. 5, pp. 630–640, 2020.

[120]. J. Hemalatha, S. A. Roseline, S. Geetha, S. Kadry, and R. Damaˇsevicius, "An efficient densenet-based deep learning ˇmodel for malware detection," Entropy, vol. 23, no. 3, p. 344, 2021.

[121]. S. Baek, J. Jeon, B. Jeong, and Y. S. Jeong, "Two-stage hybrid malware detection using deep learning," Humancentric Computing and Information Sciences, vol. 11, no. 27, pp. 10–22967, 2021.

[122]. G. Iadarola, F. Martinelli, F. Mercaldo, and A. Santone, "Towards an interpretable deep learning model for mobile malware detection and family identification," Computers & Security, vol. 105, Article ID 102198, 2021.

[123]. N. Zhang, Y. A. Tan, C. Yang, and Y. Li, "Deep learning feature exploration for android malware detection," Applied Soft Computing, vol. 102, Article ID 107069, 2021.

[124]. V. Sihag, M. Vardhan, P. Singh, G. Choudhary, and S. Son, "De-LADY: deep learning based Android malware detection using Dynamic features," J. Internet Server. Information. Security.vol. 11, no. 2, pp. 34–45, 2021.

[125]. I. Obaidat, M. Sridhar, K. M. Pham, and P. H. Phung, "Jadeite: a novel image-behavior-based approach for java malware detection using deep learning," Computers & Security, vol. 113, Article ID 102547, 2022.

[126]. J. Kim, Y. Ban, E. Ko, H. Cho, and J. H. Yi, "MAPAS: a practical deep learning-based android malware detection system," International Journal of Information Security, vol. 21, no. 4, pp. 725–738, 2022.

[127]. R. Chaganti, V. Ravi, and T. D. Pham, "Deep learning based cross architecture internet of things malware detection and classification," Computers & Security, vol. 120, Article ID 102779, 2022

[128]. X. Xing, X. Jin, H. Elahi, H. Jiang, and G. Wang, "A malware detection approach using autoencoder in deep learning," IEEE Access, vol. 10, pp. 25696–25706, 2022.

[129]. M. Kumar, "Scalable malware detection system using distributed deep learning," Cybernetics & Systems, pp. 1–29, 2022.

[130]. A. A. Hamza, I. T. Abdel Halim, M. A. Sobh, an M. Bahaa-Eldin, "HSAS-MD analyzer: a hybrid security analysis system using model-checking technique and deep learning for malware detection in IoT apps," Sensors, vol. 22, no. 3, p. 1079, 2022.

[131]. S. S Lad and A. C. Adamuthe, "Improved deep learning model for static PE files malware detection and classification," International Journal of Computer Network and Information Security, vol. 14, no. 2, pp. 14–26, 2022.

[132]. H. Cai and J. Jenkins, "Towards sustainable android malware detection," in Proceedings of the 40th International Conference on Software Engineering: Companion Proceeedings, pp. 350-351, Gothenburg Sweden, 2018, May.

[133]. E. Mariconti, L. Onwuzurike, P. Andriotis, E. De Cristofaro,G. Ross, and G. Stringhini, "Mamadroid: detecting android malware by building Markov chains of behavioral models," 2016.

[134]. L. Onwuzurike, E. Mariconti, P. Andriotis, E. D. Cristofaro, G. Ross, and G. Stringhini, "Mamadroid: detecting android malware by building Markov chains of behavioral models (extended version)," ACM Transactions on Privacy and Security (TOPS), vol. 22, no. 2, pp. 1–34, 2019.

[135]. W. Li, X. Fu, and H. Cai, "Androct: ten years of app call traces in android," in Proceedings of the 2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR), pp. 570–574, IEEE, Madrid, Spain, 2021 May.

[136]. J. Garcia, M. Hammad, and S. Malek, "Lightweight, obfuscation-resilient detection and family identification of android malware," ACM Transactions on Software Engineering and Methodology, vol. 26, no. 3, pp. 1–29, 2018.

[137]. H. Cai, N. Meng, B. Ryder, and D. Yao, "Droidcat: effective android malware detection and categorization via app-level profiling," IEEE Transactions on Information Forensics and security, vol. 14, no. 6, pp. 1455–1470, 2019. H. Gao, S. Cheng, and W. Zhang, "GDroid: android malware detection and classification with graph convolutional network," Computers & Security, vol. 106, Article ID 102264, 2021.

[138]. H. Cai, "Embracing mobile app evolution via continuous ecosystem mining and characterization," in Proceedings of the IEEE/ACM 7th International Conference on Mobile Software Engineering and Systems, pp. 31–35, 2020, July.

[139]. G. Suarez-Tangil and G. Stringhini, "Eight years of rider measurement in the android malware ecosystem: evolution and lessons learned," 2018, https://arxiv.org/abs/1801.08115.

[140]. R. Ali, S. Lee, and T. C. Chung, "Accurate multi-criteria decision making methodology for recommending machine learning algorithm," Expert Systems with Applications, vol. 71, pp. 257–278, 2017.

[141]. X. Fu and H. Cai, "On the deterioration of learning-based malware detectors for Android," in Proceedings of the 2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion), pp. 272-273, IEEE, Montreal, Canada, 2019 May.

[142]. K. Xu, Y. Li, R. Deng, K. Chen, and J. Xu, "DroidEvolver: selfevolving Android malware detection system," in Proceedings of the 2019 IEEE European Symposium on Security and Privacy (EuroSP), pp. 47–62, IEEE, Stockholm, Sweden, 2019 June.

[143]. H. Cai, "Assessing and improving malware detection sustainability through app evolution studies," ACM Transactions on Software Engineering and Methodology, vol. 29, no. 2, pp. 1–28, 2020.

[144]. M. Egele, T. Scholte, E. Kirda, and C. Kruegel, "A survey on automated dynamic malware-analysis techniques and tools," ACM Computing Surveys, vol. 44, no. 2, pp. 1–42, 2012.

[145]. M. Akour, I. Alsmadi, and M. Alazab, "(e malware detection challenge of accuracy," in Proceedings of the 2016 2nd International Conference on Open Source Software Computing (OSSCOM), pp. 1–6, IEEE, Beirut, Lebanon, 2016 December.

[146]. S. D. Nikolopoulos and I. Polenakis, "A graph-based model for malware detection and classification using system-call groups," Journal of Computer Virology and Hacking Techniques, vol. 13, no. 1, pp. 29–46, 2017.

[147]. V. Sessions and M. Valtorta, "(e effects of data quality on machine learning algorithms," ICIQ, vol. 6, pp. 485- 498, 2006. Security and Communication Networks

[148]. F. O. Catak and A. F. Yazı, "A benchmark API call dataset for windows PE malware classification," 2019.

[149]. H. Cai, X. Fu, and A. Hamou-Lhadj, "A study of run-time behavioral evolution of benign versus malicious apps in android," Information and Software Technology, vol. 122, Article ID 106291, 2020.

[150]. K. Allix, T. F. Bissyande, J. Klein, and Y. Le Traon, ´ "Androzoo: collecting millions of android apps for the research community," in Proceedings of the 2016 IEEE/ACM13th Working Conference on Mining Software Repositories (MSR), pp. 468–471, IEEE, Austin, TX, USA, 2016 May.

[151]. W. Li, X. Fu, and H. Cai, "AndroCT: ten years of app call traces in android," in Proceedings of the 2021 IEEE/ACM 18 International Conference on Mining Software Repositories (MSR), pp. 570–574, IEEE, Madrid, Spain, 2021 May.

[152]. H. Wang, J. Si, H. Li, and Y. Guo, "Rmvdroid: towards a reliable android malware dataset with app metadata," in Proceedings of the 2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR) pp. 404–408, IEEE, Montreal, QC, Canada, 2019 May.

[153]. H. Cai and B. G. Ryder, "A longitudinal study of application structure and behaviors in android," IEEE Transactions on Software Engineering, vol. 47, no. 12, pp. 2934–2955, 2021.

[154]. P. Liu, L. Li, Y. Zhao, X. Sun, and J. Grundy, "Androzooopen: collecting large-scale open source android apps for the research community," in Proceedings of the 17th International Conference on Mining Software Repositories, pp. 548–552, Republic of Korea, 2020 June.

[155]. F. Wei, Y. Li, S. Roy, X. Ou, and W. Zhou, "Deep ground truth analysis of current android malware," in Proceedings of the International Conference on Detection Of Intrusions And Malware, and Vulnerability Assessment, pp. 252–276.

[156]. Y. Zhou and X. Jiang, "Dissecting android malware: characterization and evolution," in Proceedings of the 2012 IEEE symposium on security and privacy, pp. 95–109, IEEE, San Francisco, CA, USA, 2012 May.

[157]. H. S. Anderson and P. Roth, "Ember: an open dataset for training static pe malware machine learning models," 2018, https://arxiv.org/abs/1804.04637.

[158]. R. Harang and E. M. Rudd, "SOREL-20M: a large scale benchmark dataset for malicious PE detection," 2020, https://arxiv.org/abs/2012.076

[159]. G. Kavallieratos, N. Chowdhury, S. K. Katsikas, V. Gkioulos and S. D. Wolthusen. "Threat Analysis for Smart Homes". Sep. 2019.

[160]. M. Bouzidi, N. Gupta, F. A. Cheikh, A. Shalaginov and M. Derawi. "A Novel Architectural Framework on IoT Ecosystem, Security Aspects and Mechanisms: A Comprehensive Survey". Jan. 2022.

[161]. K. Bobrovnikova, S. Lysenko, B. Savenko, P. Gaj and O. Savenko. "Technique for IoT

malware detection based on control flow graph analysis". Feb. 2022.

[162]. T. N. Phu, K. H. Dang, D. N. Quoc, N. T. Dai and N. N. Binh. "A Novel Framework to Classify Malware in MIPS Architecture-Based IoT Devices". Dec. 2019.

[163]. F. Xiao, Z. Lin, Y. Sun and Y. Ma. "Malware Detection Based on Deep Learning of Behavior Graphs". Feb. 2019.

[164]. B. Yadav and S. Tokekar. "Malware Multi-Class Classification based on Malware Visualization using a Convolutional Neural Network Model". Apr. 2023.

[165]. T. Wan et al.. "Efficient Detection and Classification of Internet-of-Things Malware Based on Byte Sequences from Executable Files". Jan. 2020.

[166]. W. Yaokumah, J. K. Appati and D. A. Kumah. "Machine Learning Methods for Detecting Internet-of-Things (IoT) Malware". Oct. 2021.

[167]. S. Hwang and J. Kim. "A Malware Distribution Simulator for the Verification of Network Threat Prevention Tools". Oct. 2021.

[168]. R. N. Alhamad and F. Alserhani. "Prediction Models to Effectively Detect Malware Patterns in the IoT Systems". Jan. 2022.

[169]. Olowoyeye O. Evaluating Open Source Malware Sandboxes with Linux malware (Doctoral dissertation, Auckland University of Technology).

[170]. Luckett P, McDonald JT, Glisson WB, Benton R, Dawson J, Doyle BA. Identifying stealth malware using CPU power consumption and learning algorithms. Journal of Computer Security. 2018 Jan 1;26(5):589-613.

[171]. Dervisis I. Linux Malware Analysis (Doctoral dissertation, University of Piraeus (Greece)).

[172]. Debnath S, Biswas S. Malware Identification, Analysis and Similarity. Cyber Security and Network Security. 2022 Mar 31:47-69.

[173]. Aslan ÖA, Samet R. A comprehensive review on malware detection approaches. IEEE access. 2020 Jan 3;8:6249-71.

[174]. Asmitha KA, Vinod P. Linux malware detection using non-parametric statistical methods. In2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI) 2014 Sep 24 (pp. 356-361). IEEE.

[175]. Landman T, Nissim N. Deep-Hook: A trusted deep learning-based framework for unknown malware detection and classification in Linux cloud environments. Neural Networks. 2021 Dec 1;144:648-85.

[176]. Cozzi E, Graziano M, Fratantonio Y, Balzarotti D. Understanding linux malware. In2018 IEEE symposium on security and privacy (SP) 2018 May 20 (pp. 161-175). IEEE.

[177]. Vurdelja I, Blaži´c I, Draškovi´c D, Nikoli´c B. Detection of linux malware using system tracers–An overview of solutions. IcEtran. 2020 Sep.

[178]. Maniriho P, Mahmood AN, Chowdhury MJ. A Survey of Recent Advances in Deep Learning Models for Detecting Malware in Desktop and Mobile Platforms. arXiv preprint arXiv:2209.03622. 2022 Sep 8.

Table 3: The models used in this study

Reference	Method	Model	Library	Platform	Dataset	Performance	DL/ML
[120]	Visualize malware as images, classify with reweighted loss	Dense CNN	Keras	Windows	Malimg, BIG 2015, MaleVis	98.46% accuracy	DL
[121]	Static op-code extraction + dynamic analysis	Bi-LTSM, CNN	Not stated	IoT	KISA 2019	Up to 95% accuracy	DL
[122]	Represent app as image, extract dex file bytes as pixels	CNN	TensorFlow, CUDA	Android	Argus Lab	97% accuracy	DL
[123]	Text classification on app analysis sequences	CNN	Keras	Android	Various datasets	96.6% accuracy	DL
[124]	Dynamic analysis logs to feature vectors	CNN with Leaky ReLU	Not stated	Android	Self-generated	98% accuracy	DL
[125]	Static analysis of Java bytecode control flow	CNN	Not stated	Java platforms	Self-generated	98.4% accuracy	DL
[126]	API call graph patterns analysis	CNN	Not stated	Android	Playstore , VirusShare	93.2% accuracy	DL
[127]	Static/dynamic analysis on ELF binaries	Bi-GRU- CNN	Keras, TensorFlow, scikitlearn	IoT	Various sources	98% detect accuracy, 100classify accuracy	DL
[128]	Bytecode to images, autoencoder reconstruction error	Autoencoder + CNN	TensorFlow	Android	Playstore , VirusShare	96.2% accuracy	DL
[129]	Distributed model, static + dynamic analysis	CNN-BiLSTM	Not stated	Windows	Various sources	97% accuracy	DL
[130]	Source code conversion, model checking	CNN	PyTorch	IoT	Not stated	95% accuracy	DL
[131]	Static PE file feature extraction	Not stated	Keras	Windows	EMBER	97.5% accuracy	ML
[132]	Visualize malware as	CNN (VGG16)	Not stated	Windows	VirusSign dataset	94.7% accuracy	ML

	RGB images, hybrid analysis						
[150]	Byteplot image visualization + CNN	CNN (Inception)	TensorFlow	Windows	EMBE R	98.6% accuracy	ML
[151]	API call graph analysis + random forest	Random forest	scikit-learn	Android	Conta gio/VirusShare	97.2% F1 score	ML
[152]	Control flow graph + SVM	SVM (RBF kernel)	LibSVM	Linux	Self-generated	95.3% accuracy	ML
[153]	Function call monitoring + isolation forest	Isolation forest	scikit-learn	IoT	IoT-23	99.1% detection rate	ML
[154]	Network traffic + naïve Bayes	Naïve Bayes	scikit-learn	IoT	ISOT	91.7% accuracy	ML
[155]	Bytecode image + ensemble of multiple models	Ensemble learner	scikit-learn	Android	Drebin	93.8% accuracy	ML
[156]	Binary execution traces + gradient boosting	Gradient boosting	XGBoost	Windows	BIG 2015	97.2% accuracy	ML
[157]	Control flow graph + CART decision tree	CART	Scikit-learn	Linux	self Mining	90.1% accuracy	ML
[158]	API call monitoring + logistic regression	Logistic regression	Scikit-learn	Android	AMD	92.4% accuracy	ML
[159]	Byteplot visualization + random forest	Random forest	Scikit-learn	Windows	VirusShare	96.7% accuracy	ML