A SURVEY OF DEEP LEARNING MODELS FOR DETECTION AND CLASSIFICATION OF POLYMORPHIC MALWARES OVER ENCRYPTED NETWORKS

Supported by tetfund
TERTIARY EDUCATION TRUST FUND

[1]Ogba Paul, [2]Moses Timothy
[1]Department of Computer Science, Federal University of Lafia, Nasarawa State, Nigeria,
[2]Department of Computer Science, Federal University of Lafia, Nasarawa State, Nigeria,
Corresponding author: moses.timothy@science.fulafia.edu.ng, ogbapaul@gmail.com

Received: September 5, 2025, Accepted: November 28, 2025

Abstract

Today, malware is among the most common and harmful kinds of cyberattacks. It affects millions of devices globally and can execute a variety of damaging operations including file encryption, confidential information extraction, system performance degradation, and so many others. The most dangerous among the malware variants is polymorphic malware. Polymorphic malware differs from simple encryption in that it can use infinite encryption techniques and modifies a portion of the decryption code with each iteration. Different malicious executions by the malware might be classified under encryption operations, depending on the type of malware. The encrypted virus usually contains an embedded transformation engine, which produces a new encryption algorithm at random. The developed algorithm integrates a new decryption key into the engine, leading to the creation of encrypted malware. Deep Learning (DL) has emerged as a promising technology for identifying such threats. Given the increasing complexity of malware variants across various network environments, DL techniques offer robust solutions for developing scalable and sophisticated models capable of detecting and classifying malware effectively, especially due to their ability to process huge volumes of data. This survey outlines the various research and efforts in application of deep learning models for polymorphic malware over encrypted networks. The survey employs systematic approach where research works and findings are presented highlighting major contributions to knowledge and key challenges and recommendations for future studies.

Keywords: Malware detection, Deep learning, polymorphic Malware, encrypted network, Decryptors.

1.0 Introduction

Malware is becoming a more sophisticated weapon used by cyber attackers, spreading faster and spreading throughout the network (Djenna et al. 2023).. Also, because modern malware may avoid detection, hinder digital forensics investigations from happening nearly instantly, and have major, far-reaching impacts due to advanced evasion strategies, it is also one of the most harmful types of cybercrime. Because of this, prompt and reliable detection is essential for efficient analysis. Malware refers to malicious software designed to damage, disrupt, or compromise the normal operation of computer systems. This term comprises a wide range of harmful programs, such as viruses, Trojan horses, spyware, adware, and other types of malicious code (Kumar 2020).. A malware detection system aims to identify malicious software. However, malware creators employ obfuscation techniques to alter the appearance of their malware, thereby fooling virus scanners that rely on pattern matching. These scanners typically focus on the syntax rather than the semantics of instructions, making them vulnerable to such variations (Le et al. 2018). Today, most malware exhibits metamorphic and polymorphic characteristics, allowing it to alter its code during spreading.

Polymorphic malware is designed to change appearance each time it replicates while maintaining the status of the original code. Polymorphic malware differs from simple encryption in that it can use infinite encryption techniques and modifies a portion of the decryption code with each iteration. Different malicious executions by the malware might be classified under encryption operations, depending on the type of malware. The encrypted virus usually contains an embedded transformation engine. This engine always produces a new encryption algorithm at random. Next, the generated algorithm links a new decryption key to the engine and encrypted malware (Galteland and Gjøsteen 2017). Polymorphic malware has an endless number of new decryptors, which makes it more difficult to identify. A major feature of this approach is the continuous code updates that accompany every new version (Aboaoja et al. 2022). Despite modifications, these malware versions can display distinct behavioral traits. Machine learning techniques can leverage on these behavioral patterns to identify previously unknown malware, a task that traditional detection methods struggle to accomplish (Gandotra, Bansal and Sofat 2014)..

Attackers are continuously developing malware that is polymorphic, giving it the ability to alter its source code as it spreads (Suraneni 2022). Machine Learning (ML) methods can therefore be utilized to detect novel viruses and categorize them into established families by analyzing their behavioral patterns using either static or dynamic analysis. Furthermore, numerous threats such as malicious software payloads, fraudulent schemes to obtain sensitive information, and unauthorized access to vital data persists; concealed inside layers of encrypted internet traffic. The 2022 report on the status of encrypted attacks analyzed 24 billion attacks from October 2021 to September 2022 for further research (Desai, 2023).. The report sheds light on the level of threats embedded within HTTPS traffic, encompassing SSL and TLS. The data reveals a consistent

upward trend in the proportion of attacks employing encrypted channels, rising from 57% in 2020 to 80% in 2021, and ultimately surpassing 85% in 2022 (Desai, 2023).

Although encryption provides important privacy and data security advantages, malware developers may use it to obscure their activities from discovery. Because firewalls and antivirus programs are unable to examine the contents of encrypted data streams, encrypted traffic presents a serious problem for them. Numerous malware types are made to function within encrypted traffic, using sophisticated techniques that make them difficult to detect and analyze and have the potential to cause serious harm. Malware uses a particularly complex technique called polymorphism, which enables it to alter its identity with every attack, making detection and mitigation even more difficult. Deep learning technique has emerged as a promising technology for detecting these polymorphic malwares. This study therefore examined different works and efforts made by researchers in detecting polymorphic malwares over encrypted network using deep learning models for polymorphic malware over encrypted networks. The survey presents major contributions made by different researchers and key challenges and recommendation for future studies.

Studies on all malware detection, identification, and classification variants using Machine Learning, Deep Learning, and metaheuristics have been examined in recently. A significant advancement in AI has been made possible by deep learning models' capacity to identify patterns in data and codes to deduce meaningful information. This power has been harnessed in various fields to protect critical systems from harm, especially malware. Irrespective of whether the network is encrypted or not, deep learning models can learn patterns from data generated and predict useful results. Several works have been in this area. For instance, Scarbrough (2021) categorized encrypted network traffic into benign and harmful types using a feature analysis. The research focused on machine learning models for feature analysis, bypassing the need for human expertise to identify relevant features in encrypted traffic. The authors trained and tested their models using three machine learning algorithms: SVM, XGBoost, and RF. They employed recursive feature elimination (RFE) to analyze features in each case. XGBoost slightly outperformed RF, achieving closely 99% accuracy, while SVM showed lower accuracy in comparison.

Zheng et al. (2020) focused on examining the SSL/TLS handshake's unencrypted sections in addition to Open Source Intelligence (OSINT) information about IP addresses and domain names. This dataset was processed using three machine learning algorithms: SVM, Autoencoder Neural Network and One-Class SVM (OC-SVM). The method employed dataset that is not balanced, with the OC-SVM model forming better than the other algorithms in terms of accuracy for malware detection.

Luo, Xu and Liu (2021) introduced a method for identifying malicious TLS traffic by concentrating on communication channel features. They developed a unique set of features based on the consistency, distribution and statistical properties of Transport Layer Security (TLS) network traffic. The researchers then employed the Random Forest (RF) algorithm to test and train their model of detection, achieving an accuracy of 97.44%. This method proved particularly effective for identifying obfuscation of malicious traffic.

Cho et al. (2020) introduced a highly efficient end-to-end deep learning model for malware detection, utilizing 2D image-based analysis. To preserve bit-level information, they improved the convolutional architecture by integrating a black-and-white embedding. The model was assessed using datasets from the Microsoft Malware Classification Challenge (BIG 2015) on Kaggle and the KISA Malware Challenge 2018. With an accuracy of approximately 97%, their method surpassed popular deep learning models for image recognition, including VGG and ResNet, in both training and testing phases, as confirmed by experimental findings.

Gómez et al. (2023) in their study employed a distributed and automated machine learning approach to detect malware within encrypted network traffic. They utilized contextual flow data and TLS metadata extracted from network streams. To evaluate the detection capabilities, three machine learning algorithms—Random Forest (RF), XGBoost, and Support Vector Machine (SVM)—were applied. Experimental results indicated that the RF classifier delivered the highest performance among the tested models. A multiclass classification framework was implemented in their approach.

In separate research, Dai et al. (2019) used effective feature extraction technique based on structural correlation to identify threats in maliciously enciphered TLS communication. An RF-based machine learning model is fed the retrieved features. The accuracy of the model was shown to be 99.38% by the experiment's outcomes.

Apart from this, Gómez et al. (2023) introduced a new unsupervised technique for identifying and grouping malicious TLS traffic. This is to ascertain whether a particular traffic is infected with malware. The researchers developed an unsupervised detection system that calculates the distance between the cluster. Also, they used 35M TLS flows and 972k traces from a commercial sandbox to test their methodology. Over network traffic, the proposed unsupervised system achieves an FDR of 0.032% and an F1 score of 0.91.

The previously mentioned machine learning methods were unable to handle end-to-end encrypted traffic classification since they depended on manually developed network traffic features. Researchers are currently turning to deep learning algorithms to create detection models that can better categorize encrypted traffic in order to overcome this issue.

Zaharin and Shariff (2021) conducted a study to differentiate malware from regular software by analyzing system calls

made by malware installed on a platform. Their approach incorporated the Debian operating system and the DRAKVUF framework. The method involved logging system calls and identifying malicious behavior through a classification model to determine whether the software was malware or legitimate. Using a self-constructed dataset, they achieved an impressive accuracy rate of 95.6%.

In their research, Khan and Ullah (2022) introduced a Deep Squeezed-Boosted and Ensemble Learning (DSBEL), a revolutionary malware detection methodology over encrypted traffic. This system uses a recently developed Squeezed-Boosted Boundary-Region Split-Transform-Merge (SB-BR-STM) CNN in conjunction with ensemble learning. Multi-path dilated convolutional, border, and regional operations are used by the suggested STM block to catch both consistent and varied worldwide malevolent patterns. Additionally, it relies on transfer learning and multi-path squeezing and boosting techniques at the initial and final stages to detect subtle pattern variations and generate diverse feature maps. The SB-BR-STM CNN's discriminative features are supplied into ensemble classifiers, such as SVM, MLP, and AdabooSTM1, to improve hybrid learning generalization. The DSBEL framework and the SB-BR-STM CNN were evaluated using the IOT_Malware dataset, showing 98.50% accuracy and effectiveness when compared to current approaches.

Another different research by Chaganti, Ravi and Pham (2022) suggested the EfficientNetB1 neural network as a way to group malware families by using a byte-level image of malware samples. They investigated how well different pretrained convolutional neural networks (CNNs) worked to find the best architecture for finding malware. They did this by focusing on how to make training and testing less computationally intensive. Also, the study looked at how these pretrained models compared to each other using different picture representation methods that had different widths. Their results showed that EfficientNetB1 did an amazing job of classifying malware with 99% accuracy on the Microsoft Malware Classification Challenge (MMCC).

In a different study, Berrueta et al. (2022 introduced a method for detecting ransomware infections by analyzing file-sharing traffic. The approach uses machine learning to identify patterns in the traffic that indicate ransomware activity, such as reading and overwriting files. It monitors communication between clients and file servers and is designed to work with both encrypted and clear text file-sharing protocols. Three machine learning models are evaluated, and the most effective one is selected for validation. The detection algorithm is trained and tested on over 2,500 hours of legitimate, non-infected user activity and more than 70 ransomware binaries from 26 different strains. The results demonstrate that the proposed method can detect all ransomware binaries, including those not encountered during training (unseen) by 100%. The study further validates the technique by analyzing the false positive rate and measuring how much user data the ransomware can encrypt before being detected.

In their study, Mane et al. (2022) employed the MalImg dataset to create an ensemble-based method for finding malware. A standard convolutional neural network (CNN) was used to extract features. An unsupervised learning model then used the K-nearest neighbor (KNN) technique to sort the malware samples into their correct categories based on the attributes that were acquired. This CNN-KNN framework was made to be simpler and worked better than other models like VGG16 and ResNet50. The suggested strategy has a high detection accuracy of 99.63% and also cut down on training time. A CNN-based technique for finding ransomware was also introduced (Marsh and Haddadpajouh 2022).

Saeed and Akram (2022) proposed a highly effective method for malware classification by leveraging different Convolutional Neural Network (CNN) designs. Their approach utilized both custom-designed CNNs and established models like ResNet-50, AlexNet, VGG-16, and Inceptionv3. Using the Malimg dataset, which contains malware images created from malware binaries, they demonstrated the effectiveness of their approach. The trained models achieved a test accuracy of 98.90%, highlighting their capability to classify malware with exceptional precision.

Tayyab et al. (2022) highlighted that malware developers have adopted evasion techniques to hide structural changes and avoid detection. In response, researchers leveraged artificial intelligence to address these tactics. They developed a convolutional neural network (CNN) that classifies PDFs as either benign or malicious based on byte-level data, using a dataset of approximately 21,000 files. This method eliminates the need for manual feature extraction or human intervention. Their approach achieved an impressive 97% accuracy in detecting malicious PDFs.

Selamat and Fakariah (2022) developed a supervised machine learning-based model for detecting polymorphic malware in unencrypted networks with data collected from open-source repositories. The findings from this research showed a detection accuracy of over 90%, but because it's an ML-based model, it has an overhead of feature engineering requirements.

Foudy (2023) researched polymorphic malware detection by using the deep Learning method. The study conducted a comparative analysis of 1D CNN, RNN, and GAN networks. Findings from the research showed that GAN outperformed both CNN and RNN by a significant performance metric. The decrease in the performance of both 1D CNN and RNN is due to these models' specialization in time series-based datasets and poor performance in either bitwise or Opcode datasets.

In a related study, Altaiy, Yıldız and Bahadır (2023 used deep learning techniques LSTM, CNN, and DNN models—to detect malware in an unencrypted network with a 96% accuracy rate. The models developed in this research use the CTU-13 dataset, which is unstandardized and unencrypted.

Further, a study by Wang and Thing (2023) on Machine Learning for Encrypted Malicious Traffic Detection using the CICIDS-2017 dataset. The performance of this model was high, achieving an accuracy of 99.5%.

Singh and Singh (2023) proposes the use of deep learning (DL) techniques to detect malware in encrypted traffic without the need for decryption. In addition to protecting user privacy, this architecture improves the effectiveness of threat detection in encrypted network traffic. Using the CTU-13 malware dataset, which includes network traffic features. Three deep learning techniques were assessed: long short-term memory (LSTM), multilayer perception (MLP), and 1D convolutional neural network (1-D CNN). The experimental results show that the testing accuracy of LSTM stabilizes at 97.33% with classification accuracy of 98.99%.

Sharma, Sharma and Kalia (2022) presented a study employing deep convolutional neural networks (CNNs) to find and classify malware that targets both Windows and IoT systems. The method used both classical learning and transfer learning to change malware binaries into grayscale, RGB, and Markov pictures. They used a Markov probability matrix to create Markov images that kept the global statistical features of the binary data, which are usually lost when images are converted. For the classification task, the study used a custom-built deep CNN and a pretrained Xception model. The Xception model was first trained on 1.5 million photos from the ImageNet dataset and then fine-tuned to look for malware images. In addition. Gabor filters to get characteristics based on texture from the pictures.

Ashawa et al. (2024) showed a better way to classify malware based on images that uses convolutional neural networks (CNNs) with ResNet-152 and the Vision Transformer (ViT). They compare the classification performance of these two architectures. A dataset consisting of 9,861 malicious executables and 6,137 benign files is converted from text files to unsigned integers and then into images. ResNet-152 processes pixel values by converting them to floating-point numbers for classification, while ViT directly analyzes the unsigned integers as pixel values. The model achieves an impressive accuracy of 99.62%. The findings suggest that the proposed system is well-suited for dynamic and complex malware environments, effectively identifying and categorizing new malware samples with a computational efficiency of 47.2 seconds.

Sato et al. (2023 explored the effectiveness of malware attacks through source code obfuscation and examined potential defense strategies. Their study made two key contributions. First, it highlighted the use of Obfuscator-LLVM (OLLVM) as a code obfuscation technique to bypass malware image classification systems. In every case, the VGG16-based image classifier failed to correctly identify malware binaries obfuscated using OLLVM. Second, the research demonstrated that this attack could be mitigated by incorporating obfuscated samples into the training process. They confirmed that a malware image classifier trained with OLLVM-generated obfuscated samples achieved 100% accuracy in classification.

Gutierrez et al. (2024) conducted a study that utilized convolutional neural networks for visual analysis and recurrent neural networks for sequential analysis. Their approach integrated static features extracted from the code with dynamic features derived from runtime behavior. The researchers used datasets from both public and private sources, including the VirusShare and Drebin databases. This comprehensive feature integration enhanced their model's ability to detect both novel and existing types of malwares. The results demonstrated that their model significantly outperformed traditional methods, which typically achieve precision rates of 88% to 89%, by achieving a precision of 98%, a recall of 97%, and an F1-score of 0.975. Furthermore, their model surpassed other deep learning methods referenced in their study, reaching precision levels of up to 96%. The summary of related works is given in table 1.

1.1 Motivation

The motivation for this survey stems from the growing sophistication of malware and the limitations of existing detection methods. Traditional approaches struggle to keep pace with the dynamic nature of polymorphic malware, especially when transmitted over encrypted networks. Deep learning offers a potential solution, but the field is rapidly evolving, and a comprehensive survey is needed to consolidate existing research, identify gaps, and guide future work.

1.2 Research Objective

The research into detection and classification of all varieties of malware across different kinds of networks is still an area of active research. The objectives of this survey are to

1. Conduct an in-depth survey into recent efforts by researchers in the area of detection and classification of polymorphic malware across different types of networks.
2. Highlight the performance and recent trends in the application of deep learning for polymorphic malware detection, classification and prevention.
3. Highlight various research gaps which will serve as a baseline for future studies.
4. Provide a platform for the development of state-of-the-art techniques in tackling polymorphic malware on critical, encrypted and real-time networks.
5. Suggest future research directions.

1.3 Survey Scope

This survey focuses on deep learning techniques for polymorphic malware detection and classification, with an emphasis on encrypted network environments. It covers studies published in reputable journals in the last decade, including supervised, unsupervised, and hybrid deep learning models.

2.0 Survey Strategy and Selection Criteria

This survey employs systematic approach in which articles are carefully reviewed and selected after certifying that they fulfil certain required criteria. This review focuses on peer-reviewed articles, conference papers, and preprints published between 2018 and 2025. Databases such as IEEE Xplore, ResearchGate, ACM Digital Library, SpringerLink, and Google Scholar were searched using keywords like "deep learning," "polymorphic malware," "encrypted traffic," and "malware classification." Studies were included if they proposed DL-based methods for detecting or classifying polymorphic malware in encrypted network traffic. Similarly, articles not satisfying these criteria are screened out. A summary of methodology employed is shown in Figure 1 below.

2.1 Inclusion and Exclusion Criteria

Inclusion Criteria: Survey focusing on deep learning-based detection and classification of polymorphic malware in encrypted network environments.

Exclusion Criteria: Studies not related to deep learning, non-encrypted network scenarios, or non-polymorphic malware.

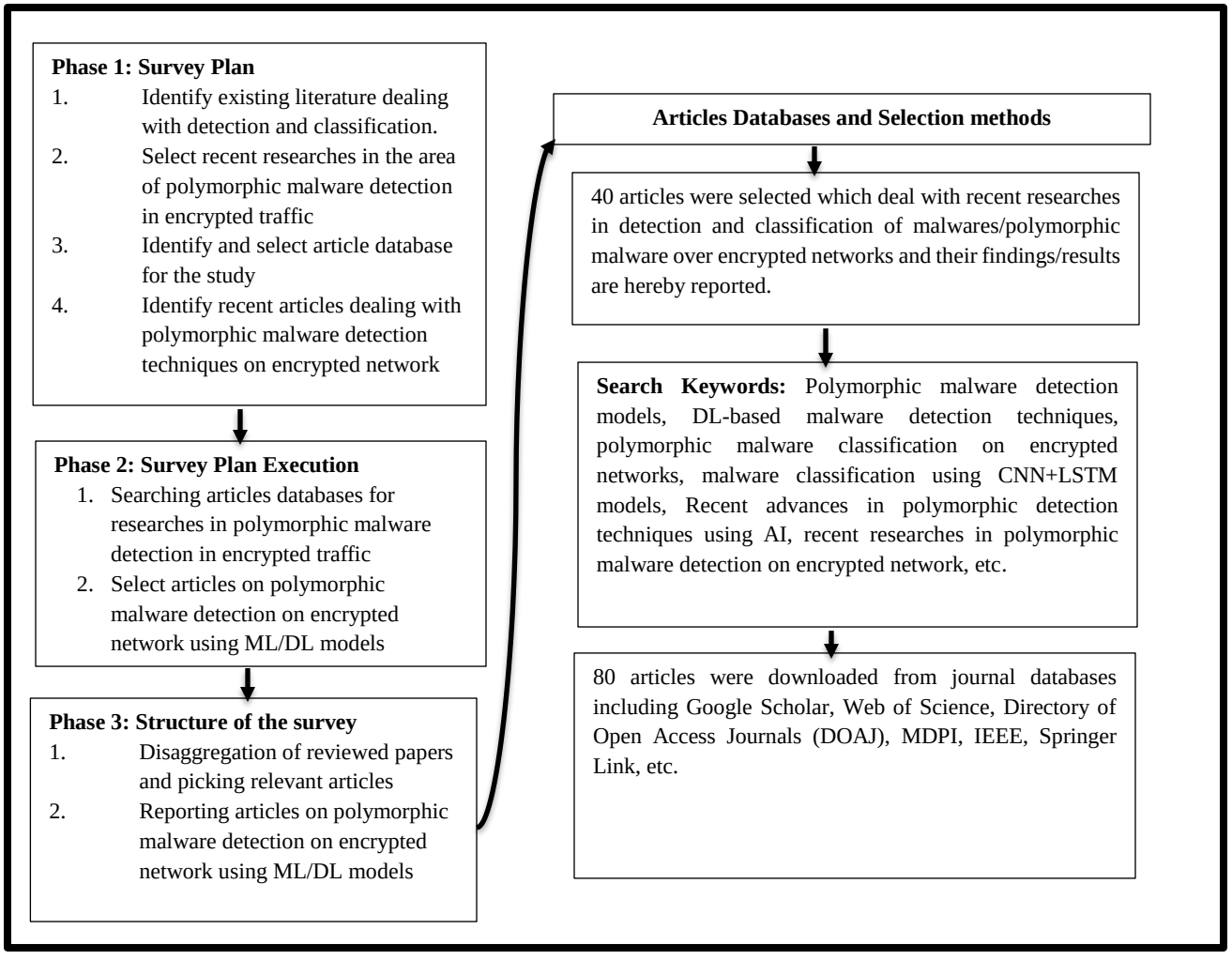

Figure 1: Survey Strategy

Here is

Table 1: Summary of Literature Review

S/No.	Authors (Date of Publication)	ML/DL Techniques adopted	Dataset Used	Highest Performance Accuracy	Strengths	Weaknesses
1	(Marcedo and Santone 2020)	BiLSTM	Android bytecode	98%	The deep learning model employed (BiLSTM) has proven to be a good classifier for all categories of dataset	Only the static behavior of network traffic was considered
2	(Obaidat et al. 2022)	CNN	Extracted images, bytecode	98.4%	They used a deep Convolutional Neural Network (CNN), a promising classifier combined with an object detection algorithm.	They did not consider the obfuscation nature of the malware
3	(Anderson and McGrew 2017)	ML, SVM, RF, NB, KNN	Private dataset encrypted	94%	Used an ensemble technique that reduces misclassification errors	The dataset used was not a validated benchmark for deep learning framework.
4	(Hamad, Durad and Yousaf 2018)	Counted by BN IDS LOOL XGBoost	Not specified	98.5%	High-performance accuracy	ML-based techniques require so many feature extraction overheads
5	(Dai et al. 2019)	RF, SVM, DT, XGBOOST	CTU-13	97.70%	The use of multi-view features of the network traffic is an excellent and rich feature or pattern approach that guarantees better model performance	ML-based techniques require so many feature extraction overheads
7	(Luo, Xu and Liu 2021)	RF	Privately developed network dataset	97.44%	Effective performance results are guaranteed when the model is trained and evaluated using the consistency, distribution, and statistical features of TLS network traffic.	The dataset is not benchmarked, and the RF classifier alone is not good for larger dataset
9	(Gómez et al. 2023)	Clustering method	Commercial Sandbox	91%	They considered TLS payload inspection and used a sandbox, which is suitable for self-test and experiment	Requires manual building of network traffic features and is unable to handle detection and classification of end-to-end enciphered data.
11	(Selamat and Fakariah 2022)	RF, SVM, KNN, NB	Open source data	98%	Ensemble models used to guarantee high-performance measures	The dataset used is not benchmarked; thus, the reported performance needs to be verified further.
12	(Foudy 2023)	1CNN, RNN, GAN	Synthetic dataset	97%	The ensemble of deep learning models for classification reduces feature engineering requirements.	The use of synthetic datasets for model training and validation suffers from generalization issues

13	(Altaiy, Yıldız and Bahadır 2023)	LSTM, CNN, DNN	CTU-13 Dataset	96%	Achieved high-performance measures due to selected approach and models	The models developed in this research used the CTU-13 dataset, which is unstandardized and unencrypted
14	(Wang and Thing 2023)	SVM, RF, NBS`	CICIDS-2017 dataset	99.5%	High-performance metric was reported	The ML-based model adds high computational overheads due to feature manipulations
15	(Singh and Singh 2023)	1DCNN, LSTM, MLP	CTU-13 Dataset	99.10%	High-performance metrics reported due to the implementation of the ensembled models	The models developed in this research used the CTU-13 dataset, which is unstandardized and unencrypted
16	(Ali and Singh 2023)	SVM, RF, VGBOOST	Open-source dataset encrypted	99%	High-performance metrics reported due to the implementation of the ensembled models	The dataset used was unstandardized; thus, the reported metrics need further verifications.
17	(Ashawa et al. 2024)	CNN, ResNet-152 and vision transformer (ViT)	Private dataset	99.62%	capable of being implemented in dynamic and complex malware environments	Utilized few malware sample which could lead to over performance
18	(Sharma, Sharma and Kalia 2022)	CNN	Windows malware image dataset (Kaggle)	Not stated	They utilized large number of datasets	Too much Overhead feature engineering and use of classification enhancer
19	(Rezende et al. 2018)	CNN and VGG16	ImageNet	92.97%	They utilized large volume of dataset with different categories	Focus was on VGG16 Enhancer bottleneck with much Overhead feature engineering
20	(Zaharin and Shariff 2021)	ML	self-generated dataset	95.6%	They considered static, dynamic and hybrid method of malware detection	Self-generated dataset which is unstandardized
22	(Khan and Ullah 2022)	CNN, SVM, MLP and AdabooSTM1	Simulated dataset	98.50%	Suitable for early detection of malware	Overfitting, Limited Interpretability and Computational Overhead due to too much transfer learning
23	(Tayyab et al. 2022)	CNN	virusSign, Contagio dump, VirusShare, and VirusTotal	97%	Suitable for PDF malware attachments with obfuscation techniques	PDF file related malware detection only
24	(Berrueta et al. 2022)	ML	Self-generated	100%	Suitable to monitor client and server traffic in encrypted traffic	Polymorphic malware not considered, and their claim cannot be generalized
25	(Gutierrez et al. 2024)	CNN and RNN	Public and private dataset	98%	They used comprehensive public and private dataset	Polymorphic malware was not considered
26	(Cho et al. 2020)	2D CNN, VGG and ResNet	KISA Malware Challenge 2018 dataset	97%	specialized in handling binary image data	Few datasets which could lead to over performance of model
27	(Sato et al. 2023)	CNN and VGG-16	Simulated dataset	100%.	Effective in detection of code obfuscation	Over performance due to unstandardized dataset

28	(Saeed and Akram 2022)	CNN, AlexNet, VGG-16 and ResNet-50	Public Malimg dataset from Kaggle	98.90%	Rapid detection of malware	polymorphic malware not considered
29	(Chaganti, Ravi and Pham 2022)	CNN	Open source MMCC dataset	99%	Detection of obfuscating malware in real time	Encrypted traffic not considered

4.0

Findings, Challenges and Future Directions

4.1 Findings

The survey of existing literature reveals several key findings regarding the methodologies, data sources, and performance of DL models in this domain.

4.1.1 Dominant Deep Learning Architectures

Research has converged on a few primary DL architectures, often used in combination:

Convolutional Neural Networks (CNNs): Excell at extracting spatial features from data. They are primarily used in two ways:

Image-based Classification: Raw byte sequences from malware binaries or network packets are converted into grayscale images. CNNs then learn to identify visual patterns indicative of malware, which is effective against polymorphism as the core functionality often leaves a recognizable visual footprint (Singh and Singh 2023; Ali and Singh 2023). A typical workflow is shown in Figure 1.

1D-CNNs for Sequential Data: Applied directly to sequences of packet sizes, inter-arrival times, or byte streams to automatically extract relevant features without manual engineering (Wang and Thing 2023).

Recurrent Neural Networks (RNNs) / Long Short-Term Memory (LSTM) Networks: Ideal for capturing temporal dependencies and sequences in network traffic. They model the flow of a network session, learning patterns in the order and timing of packets, which is crucial for identifying command-and-control (C2) beaconing or exfiltration patterns within encrypted flows (Altaiy, Yıldız and Bahadır 2023). Hybrid Models (CNN-LSTM): The most promising approach combines the strengths of both architectures. CNNs first extract high-level features from short sequences or packet bursts, and LSTMs then model the long-term dependencies between these features across the entire network flow (Gómez et al. 2023; Dai et al. 2019). A standard hybrid framework is depicted in Figure 2.

Autoencoders (AEs) and Variational Autoencoders (VAEs): Used for unsupervised and semi-supervised anomaly detection. They learn a compressed representation of "normal" benign traffic. Polymorphic malware traffic, which deviates from this learned norm, results in a high reconstruction error, flagging it as anomalous (Anderson and McGrew 2017).

4.1.2 Critical Feature Sets for Encrypted Traffic Analysis

Since payload inspection is impossible, models rely on features extracted from the metadata of encrypted traffic:

Packet Size and Sequence: The sizes of packets (e.g., first N packets, average packet size) and their sequence form a unique fingerprint for many malware families.

Inter-Arrival Times: The timing between packets can reveal patterns like periodic beaconing.

TLS/SSL Handshake Features: Features extracted from the unencrypted handshake phase are highly discriminative. These include cipher suites offered, TLS extensions, elliptic curve groups, order of parameters, and certificate information (e.g., validity period, issuer) (Dai et al. 2019; Hamad, Durad and Yousaf 2018).

Flow Statistics: Features like total bytes, packets, duration, and average packets per second.

Byte-Level Features: The raw byte stream of the initial packets of a flow (before encryption is fully established) can be processed directly by 1D-CNNs.

4.1.3. Performance Superiority

The consensus across studies is that DL models, particularly hybrid CNN-LSTM architectures, significantly outperform traditional machine learning models (e.g., Random Forests, SVM) and signature-based tools. They achieve higher detection rates (recall) and lower false positive rates (FPR) on benchmark datasets when trained on sufficiently large and diverse data, demonstrating a robust ability to generalize across polymorphic variants (Gómez et al. 2023; Dai et al. 2019).

4.2 Challenges

Despite promising results, the deployment of these systems in production environments faces formidable challenges.

4.2.1 Data Availability and Quality

Labeled Datasets: Acquiring large-scale, high-quality, labeled datasets of encrypted malware traffic is difficult. Public datasets (e.g., CICAndMal2017, CICDarknet2020, Stratosphere IPS) become outdated quickly.

Class Imbalance: Real-world network traffic is overwhelmingly benign, leading to severely imbalanced datasets that can bias models toward the majority class.

Data Privacy: Capturing live encrypted traffic from real networks raises significant privacy concerns and legal hurdles.

4.2.2 Adversarial Evasion (Adversarial Attacks)

DL models are vulnerable to adversarial examples—carefully perturbed inputs designed to cause misclassification. In this context, an adversary can subtly modify traffic patterns to evade detection (Rigaki and Garcia 2018).

Packet Manipulation: Adding dummy packets, splitting packets, or perturbing inter-arrival times.

TLS Fingerprinting Evasion: Mimicking the TLS fingerprints of popular browsers or applications.

These attacks are a critical threat to the reliability of DL-based detectors.

4.2.3 Computational Complexity and Real-Time Processing

Deep learning models, especially large CNNs and RNNs, are computationally intensive. Performing deep packet inspection on high-speed network links (e.g., 100 Gbps+) in real-time requires significant hardware acceleration (e.g., GPUs, TPUs), increasing the cost and complexity of deployment.

4.2.4 Model Interpretability and Explainability (XAI)

DL models are often "black boxes." It is difficult for security analysts to understand why a particular flow was classified as malicious. Without explanations or actionable intelligence (e.g., "this flow is malicious because it exhibits periodic beaconing every 5 seconds"), analysts cannot verify the model's decision or use its output for threat hunting and response (Sherry et al. 2015).

4.2.5 Generalization and Concept Drift

Malware authors constantly evolve their tactics. A model trained on data from one year may become ineffective against new malware families or new versions of protocols (e.g., TLS 1.3 offers more privacy, hiding some previously useful features like SNI through encrypted SNI). This "concept drift" requires continuous retraining with fresh data.

5.0 Conclusion, Recommendation and Future Directions

5.1 Conclusions

The persistent use of encrypted networks to safeguard data privacy has posed significant problems to conventional deep packet inspection techniques. Current deep packet inspection models used for network analytics and security are becoming ineffective due to their inability to analyze encrypted traffic, due to the polymorphic nature of certain malware.

This work presents an in-depth review of advance detection and classification of polymorphic malware as well as other variants of malware using Deep learning-based models. Many researcher's models have shown great promise in detecting and classifying polymorphic malware across encrypted networks but there is need for improvement especially polymorphic malware presence because of their obfuscation. Using advanced architectures like CNNs, RNNs, and transformers, these models can efficiently analyze encrypted traffic and identify malicious behavior. However, challenges such as adversarial malwares behavior, explainability, and scalability remain issues. Future research should look at developing robust, explainable, and scalable solutions to address these challenges and improve the security of encrypted payloads.

5.2 Recommendation

One approach is to enhance extraction from encrypted traffic by emphasizing non-payload attributes such as TLS headers, certificate metadata, and flow entropy. This helps overcome the constraints of encryption. In addition to this, attention-based models like Transformers can be employed to identify malicious patterns within sequential traffic data.

Secondly, we recommend adversarial training to mitigate evasion tactics e.g., malware changing TLS fingerprints and hybrid models like CNN-LSTM for spatial and temporal analysis of network behavior to improve model robustness.

Thirdly, there is need for enhanced real-time detection by implementing lightweight models like MobileNet or applying knowledge distillation to suit edge devices. Efficiency can be further improved through hardware acceleration using GPUs or TPUs and by applying model pruning techniques.

The fourth suggestion is that there is need to include actual encrypted malware traffic e.g., CICMalDroid-2023, Stratosphere Labs datasets and simulate polymorphic versions using GANs to increase training diversity.

5.3 Future Directions

To transition from academic research to operational security systems, future work should focus on the following directions.

(a) Development of Explainable AI (XAI) Techniques

Integrating XAI (e.g., SHAP, LIME, attention mechanisms) is paramount. Models should not only classify but also highlight the specific features (e.g., "this specific TLS ciphersuite and the 2-second packet interval were the top contributors to this malicious verdict") that led to the decision. This builds trust and enables human-in-the-loop validation.

(b) 3.2. Robustness Against Adversarial Attacks

Research must focus on developing adversarially robust models for network intrusion detection. This includes:

Adversarial Training: Training models on a mixture of clean and adversarially generated traffic examples.

Defensive Distillation: Using a distilled model that is smoother and harder to attack.

Detection of Adversarial Examples: Building meta-models to distinguish between legitimate and adversarial inputs.

(c) 3.3. Self-Supervised and Semi-Supervised Learning

To mitigate the data labeling problem, future models should leverage:

Self-Supervised Learning (SSL): Models can be pre-trained on vast amounts of unlabeled network data to learn general representations of traffic flow.

Semi-Supervised Learning: Using a small amount of labeled data alongside large pools of unlabeled data to improve learning efficiency and adaptability.

(d) 3.4. Standardized Benchmarking and Datasets

The community needs continuously updated, standardized benchmark datasets that reflect modern encryption and malware trends. These benchmarks should include metrics for not only accuracy but also computational efficiency, robustness, and resilience to concept drift.

(e) 3.5. Hybrid AI-Powered Security Systems

The future lies in hybrid systems that combine the strengths of different approaches:

DL + Signature Analysis: Using DL for initial broad detection and traditional tools for confirmation on decrypted payloads where possible.

DL + Threat Intelligence: Correlating model predictions with external threat intelligence feeds for enhanced context and verification.

Federated Learning: Enabling multiple organizations to collaboratively train a model without sharing their sensitive traffic data, thus improving the model's diversity and generality while preserving privacy.

References

Aboaoja, F.A. et al. (2022) 'Malware Detection Issues, Challenges, and Future Directions: A Survey, 12(17), p. 8482.

Ali, A. and Singh, V.P. (2023) 'Machine learning

applications in Intrusion Detection Systems (IDS)', . Available at: https://www.jetir.org/view?paper=JETIR2312240 (Accessed: 2023).

Altaiy, M., Yıldız, İ. and Bahadır, B.U. (2023) 'Malware detection using deep learning algorithms', 7(1), pp. 11–26.

Anderson, B. and McGrew, D. (2017) 'Machine learning for encrypted malware traffic classification: Accounting for noisy labels and non-stationarity', pp. 1723–1732.

Ashawa, M. et al. (2024) 'Enhanced Image-Based Malware Classification Using Transformer-Based Convolutional Neural Networks (CNNs)', 13(20), p. 4081.

Berrueta, E. et al. (2022) 'Crypto-Ransomware Detection Using Machine Learning Models in File-Sharing Network Scenario With Encrypted Traffic'.

Chaganti, R., Ravi, V. and Pham, T.D. (2022) 'Image-Based Malware Representation Approach With EfficientNet Convolutional Neural Networks for Effective Malware Classification', 69, p. 103306.

Cho, M. et al. (2020) 'MAL2D: 2D Based Deep Learning Model for Malware Detection Using Black and White Binary Image', E103.D(4), pp. 896–900.

Dai, R. et al. (2019) 'SSL malicious traffic detection based on multi-view features', pp. 40–46. doi: 10.1145/3371676.3371697.

Desai, D. (2023), Zscaler.

Djenna, A., Bouridane, A., Rubab, S. and Marou, I.M. (2023) 'Artificial Intelligence-Based Malware Detection, Analysis and Mitigation', 15(3), p. 677.

Foudy, M.C. (2023) 'Detecting Polymorphic Malware Using Artificial Intelligence and Deep Learning: A Comparative Analysis of CNNs, RNNs, and GNNs. *IEEE Publication*.

Galteland, H. and Gjøsteen, K. (2017) 'Malware, Encryption, and Rerandomization - Everything Is Under Attack, pp. 233–251.

Gandotra, E., Bansal, D. and Sofat, S. (2014) 'Malware Analysis and Classification: A Survey', 5(2), pp. 56–64.

Gómez, G. et al. (2023) 'Unsupervised detection and clustering of malicious TLS flows', 2023, pp. 1–17. doi: 10.1155/2023/3676692.

Gutierrez, R. et al. (2024) 'Application of Deep Learning Models for Real-Time Automatic Malware Detection', *IEEE Access*, 12, pp. 107742–107756.

Hamad, M., Durad, M.H. and Yousaf, M. (2018) *Mitigation of the effect of standard network attacks in SSL encrypted traffic by*

Khan, S.H. and Ullah, W. (2022) 'A New Deep Boosted CNN and Ensemble Learning Based IoT Malware Detection.

Kumar, R. (2020) 'Machine Learning Applications in Malware Analysis, 5(2), pp. 56–64.

Le, T. et al. (2018) 'Obfuscation Techniques in Modern Malware, pp. 45–52.

Luo, Z., Xu, S. and Liu, X. (2021) 'Scheme for Identifying Malware Traffic With TLS Data Based on Machine Learning, 6(1), pp. 77–83.

Mane, D. et al. (2022) 'An Adaptable Ensemble Architecture for Malware Detection., pp. 647–659.

Marcedo, M. and Santone, A. (2020) 'A novel image-behavior-based approach for Java malware detection using deep learning, 113(C).

Marsh, K. and Haddadpajouh, H. (2022) 'Ransomware Threat Detection: A Deep Learning Approach, pp. 253–269.

Obaidat, I., Sridhar, M., Pham, K.M. and Phung, P.H. (2022) 'Jadeite: A novel image-behavior-based approach for Java malware detection using deep learning, 113, p. 102547. doi: 10.1016/j.cose.2021.102547.

Rezende, E. et al. (2018) 'Malicious software classification using VGG16 deep neural network's bottleneck features, pp. 51–59. doi: 10.1007/978-3-319-77028-4_9.

Rigaki, R. and Garcia, S. (2018) 'Bringing a GAN to a Knife-Fight: Adapting Malware Communication to Avoid Detection, pp. 70–75.

Saeed, T. and Akram, A. (2022) 'DeepMalware: A Deep Learning Based Malware Images Classification. *of the IEEE Conf*

Sato, H. et al. (2023) 'Plain Source Code Obfuscation as an Effective Attack Method on IoT Malware Image Classification, pp. 940–945.

Scarbrough, B. (2021) , White Paper.

Selamat, N.S. and Fakariah, F.H.M.A. (2022) 'Polymorphic Malware Detection Based on Supervised Machine Learning, 6(3), pp. 8538–8547.

Sharma, O., Sharma, A. and Kalia, A. (2022) 'Windows and

IoT Malware Visualization and Classification With Deep CNN and Xception CNN Using Markov Images, 60(2), pp. 349–375.

Sherry, J., Lan, C., Popa, R.A. and Ratnasamy, S. (2015) 'BlindBox: Deep Learning over encrypted inference, pp. 1–12.

Singh, A. and Singh, V.P. (2023) 'Machine learning for encrypted malicious traffic detection using the CICIDS-2017 dataset'.

Suraneni, N. (2022) 'Malware Detection and Analysis', *Culminating Experience Projects*.

Tayyab, U.H. et al. (2022) 'Identification of Malicious PDFs Using Convolutional Neural Networks, 10(3), pp. 51–57.

Wang, Z. and Thing, V.L. (2023) 'Feature mining for encrypted malicious traffic detection with deep learning and other machine learning algorithms, 128, p. 103143. doi: 10.1016/j.cose.2023.103143.

Zaharin, M.R.I.A. and Shariff, S.M. (2021) 'Malware Classification Based on System Call, pp. 387–398.

Zheng, R. et al. (2020) 'Detecting Malicious TLS Traffic Based on Communication Channel Features, pp. 14–19.