УДК 004.056.57, 004.7, 004.8

Unveiling Malicious Patterns: Autoencoder-Based Malware Detection

Baghirov E. (Institute of Information Technology, Ministry of Science and Education of Republic of Azerbaijan),

Afzal M. (National University of Computer and Emerging Science)

Malware detection poses a significant challenge in cybersecurity, particularly with the increasing sophistication of attack methods. This study introduces an autoencoder-based approach to detect malware by learning the structure of benign data and identifying anomalies through reconstruction loss. By focusing on the detection of deviations in data patterns, this method offers an effective solution for identifying both known and unknown malware. Using the MALIMG dataset, the approach is evaluated with standard metrics such as accuracy, precision, recall, and F1-score, demonstrating strong performance and computational efficiency. This work highlights the potential of autoencoders as a robust anomaly-based detection tool.

Keywords: *malware detection, autoencoder, anomaly detection, reconstruction loss, cybersecurity, malware classification, machine learning, malware analysis*

1. Introduction

Malware, short for malicious software, refers to programs intentionally designed to cause harm to computer systems, steal sensitive information, or disrupt operations [1]. The rapid evolution of malware, with increasing sophistication and diversity, has made traditional detection techniques, such as signature-based approaches, less effective. Signature-based methods rely on known patterns to identify threats, leaving systems vulnerable to novel or polymorphic malware that can easily evade detection. This has necessitated the exploration of advanced detection techniques that can identify malware based on its behavior or anomalies in data patterns [2,3].

Recent advancements in machine learning and deep learning have provided promising alternatives to traditional methods for malware detection [4]. Unlike static signature-based techniques, these methods learn patterns from large datasets, enabling the detection of both known and unknown threats [5]. Among these, anomaly-based approaches have garnered attention for their ability to identify deviations from normal system behavior. Autoencoders, in particular, are well-suited for this task as

they are designed to learn compact representations of benign data and highlight anomalies that deviate from these learned patterns.

In this study, we explore the potential of autoencoders for malware detection. By training the autoencoder exclusively on benign samples, the model learns to reconstruct normal data patterns with minimal error. Malware, being anomalous, exhibits higher reconstruction loss, allowing it to be effectively identified. This reconstruction loss serves as the core metric for distinguishing between benign and malicious samples.

The proposed method is evaluated using the MALIMG dataset [6], a well-known benchmark dataset for malware detection. Our experimental results demonstrate the effectiveness of the autoencoder in detecting malware with high accuracy and efficiency. This work also addresses key challenges such as class imbalance and model evaluation using a variety of performance metrics, including accuracy, precision, recall, and F1-score.

The remainder of this paper is structured as follows:

- Section 2 reviews related work, highlighting existing approaches and gaps in malware detection research.
- Section 3 details the methodology, including dataset preparation, autoencoder architecture, and evaluation metrics.
- Section 4 presents the experimental results and analysis.
- Finally, Section 5 concludes the paper and outlines future research directions aimed at improving the precision and robustness of the proposed approach.

2. Related works

The method proposed by Xing et al. [4] employs autoencoders in conjunction with grayscale malware image representations to detect malicious software. The approach involves converting the bytecode of Android malware and benign applications into grayscale images, which serve as input for an autoencoder network. The model leverages the reconstruction error of these images to differentiate between malware and benign samples. However, the method has some limitations. The preprocessing stage of converting bytecode into grayscale images can introduce redundancy and inefficiency, potentially impacting the robustness of the model. Additionally, the reliance on grayscale image representation may restrict the applicability of the approach to certain types of malware, potentially missing important features that could be captured using other representations or techniques.

Panchagnula et al. [7] proposed a deep learning-based method for malware detection using autoencoders, where malware samples are transformed into grayscale images for feature extraction

and classification. The methodology involves two autoencoder models: AE-1, which assesses the feasibility of representing software features using grayscale images, and AE-2, which focuses on classifying malware from benign software. AE-1 uses unsupervised learning for feature extraction, while AE-2 integrates supervised learning with additional layers for classification. The model's performance was evaluated on multiple datasets containing various malware types, achieving high accuracy and F1-scores. The approach has limitations, including reliance on grayscale representations that may overlook complex malware behaviors, high computational costs for training autoencoders, and potential challenges in generalizing to unseen malware variants. These factors highlight the need for further enhancements, such as dynamic analysis or ensemble methods, to improve detection capabilities.

In a study by Halim et al. [8], the authors explored the use of recurrent neural networks (RNNs) for malware detection. By combining Long Short-Term Memory (LSTM) networks with Convolutional Neural Networks (CNNs), the model was designed to address both spatial and temporal challenges in classifying malware. This hybrid approach aimed to improve detection accuracy by capturing intricate patterns in malware behavior over time. However, despite its promising results, the model faced challenges related to high computational complexity and the need for large, labeled datasets for effective training.

In another work, MeMalDet by the authors [9] utilizes deep autoencoders for feature extraction combined with a stacked ensemble of supervised classifiers to detect malware through memory analysis, addressing limitations of traditional static and dynamic techniques. While the method achieves high accuracy (98.82%) and incorporates temporal evaluations for realistic testing, its reliance on computationally intensive memory analysis and a lack of adaptability to rapidly evolving malware behaviors remain significant limitations.

3. Methodology

This section outlines the approach employed for malware detection using autoencoders, detailing the dataset preparation, model architecture, training process, and evaluation criteria.

Dataset description. The dataset used in this study is the MALIMG dataset [6], a benchmark dataset commonly employed in malware detection research. It comprises grayscale images generated from the binary files of malware samples, representing 25 distinct malware families. These images are constructed by mapping the byte sequences of binary files into pixel values, creating visual representations that retain the structural information of the original binaries. This transformation allows machine learning models to leverage image-based analysis for malware detection.

The dataset is split into training, validation, and testing sets, with a stratified partitioning approach to preserve class distributions. During training, the autoencoder is exposed only to benign samples, enabling it to learn the structure of normal data. Malware samples are introduced during testing to evaluate the model's ability to detect anomalies based on reconstruction loss.

Table 1 provides an overview of the class distribution in the MALIMG dataset, listing all malware classes and their respective sizes.

TABLE 1. Class Distribution in the MALIMG Dataset.

Malware Class	Class Size	Malware Class	Class Size
Adialer.C	122	Lolyda.AA2	184
Agent.FYI	116	Lolyda.AA3	123
Allaple.A	2949	Lolyda.AT	159
Allaple.L	1591	Malex.gen!J	136
Alueron.gen!J	198	Obfuscator.AD	142
Autorun.K	1060	Rbot!gen	158
C2LOP.gen!g	200	Skintrim.N	80
C2LOP.P	146	Swizzor.gen!E	128
Dialplatform.B	177	Swizzor.gen!I	132
Dontovo.A	162	VB.AT	408
Fakerean	381	Wintrim.BX	97
Instantaccess	431	Yuner.A	800
Lolyda.AA1	213		

Autoencoder Architecture. An autoencoder is an unsupervised neural network designed to learn compressed representations of data by reconstructing the input as accurately as possible through an encoder-decoder architecture [7,9,10,11]. The encoder compresses input data into a latent representation, capturing its essential features, while the decoder reconstructs the original data from this compressed representation [12]. By minimizing reconstruction loss, such as Mean Squared Error, the autoencoder effectively models the structure of the input data. In malware detection, autoencoders are trained exclusively on benign samples to learn their normal patterns. During testing, malicious data, being anomalous, results in higher reconstruction loss, enabling the detection of novel or zero-day malware without relying on predefined signatures. This anomaly-based detection makes autoencoders a powerful and adaptive tool in cybersecurity. Figure 1 illustrates the architecture of an autoencoder-based anomaly detection system.

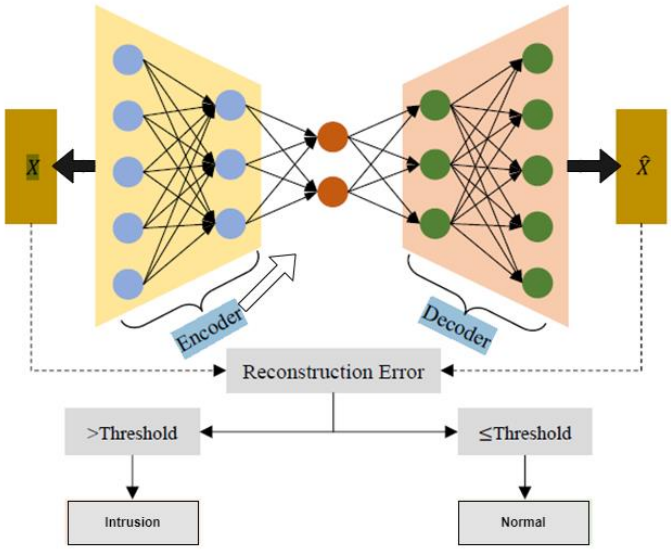

Figure 1. General architecture of autoencoder.

Evaluation metrics. To assess the performance of the proposed autoencoder-based malware detection approach, we employ a range of evaluation metrics commonly used in classification tasks. These metrics provide a comprehensive view of the model's effectiveness and its ability to generalize to unseen data. The metrics used in this study are as follows:

Accuracy: This metric measures the proportion of correctly classified samples out of the total number of samples. It is defined as:

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FN + FP)}$$

Precision: Precision evaluates the model's ability to correctly classify positive (malicious) samples, minimizing the occurrence of false positives. It is defined as:

$$Precision = \frac{TP}{(TP + FP)}$$

Recall (Sensitivity): Recall measures the model's capability to identify all positive (malicious) samples. It is particularly important in malware detection, where missing a malicious sample can have severe consequences. It is defined as:

$$Recall = \frac{TP}{(TP + FN)}$$

F1-Score: This metric provides a balance between precision and recall, especially useful when the dataset is imbalanced. It is the harmonic mean of precision and recall, defined as:

$$F1 - score = \frac{2 * precision * recall}{(precision + recall)}$$

where TP, TN, FP, and FN denote True Positives, True Negatives, False Positives, and False Negatives, respectively.

Reconstruction Error: The reconstruction error is used to classify samples as benign or malicious. It is calculated as the difference between the input and the reconstructed output. A predefined threshold is used to separate normal samples from malware.

4. Experimental results and analysis

This section presents the experimental setup, results, and analysis to evaluate the performance of the proposed autoencoder-based malware detection method, highlighting its effectiveness and limitations.

Enviromental setup. For the experiments conducted in this research, the computational setup included a Linux-based system, which was chosen for its stability and compatibility with various machine learning frameworks. The system was equipped with 32GB of RAM, providing ample memory for handling large datasets like Malimg and training complex models such as GANs and autoencoders. The GPU used was an NVIDIA RTX 3090, which offers 24GB of dedicated VRAM and is optimized for high-performance deep learning tasks, significantly accelerating the training process for both generative models and autoencoders. The CPU was an Intel Core i9-13900K, a 24-core processor with 32 threads, ensuring fast and efficient handling of the data preprocessing tasks and other CPU-bound operations. This combination of hardware components allowed for the efficient execution of all computationally intensive tasks, minimizing the training time and enabling seamless experimentation. The setup was further supported by robust software tools, including popular deep learning library i.e PyTorch, which were leveraged for model development and training.

Experiments and results. During inference, a sample is classified as malware or benign based on its reconstruction loss. We experimented with several threshold values to find the optimal threshold that maximizes classification performance. After extensive testing, a threshold of 0.95 was found to produce the best results. If the reconstruction loss of an input image is below 0.95, it is classified as malware (label 0). If the reconstruction loss is above 0.95, it is classified as benign (label 1). We evaluated our model using the test set, calculating the confusion matrix and various performance metrics, including precision, recall, and F1-score. Figure 2 shows the training and validation loss curves over 250 epochs, indicating that the model converges effectively. Figure 3 presents the confusion matrix for the binary classification task, with 0 representing malware and 1 representing benign samples.

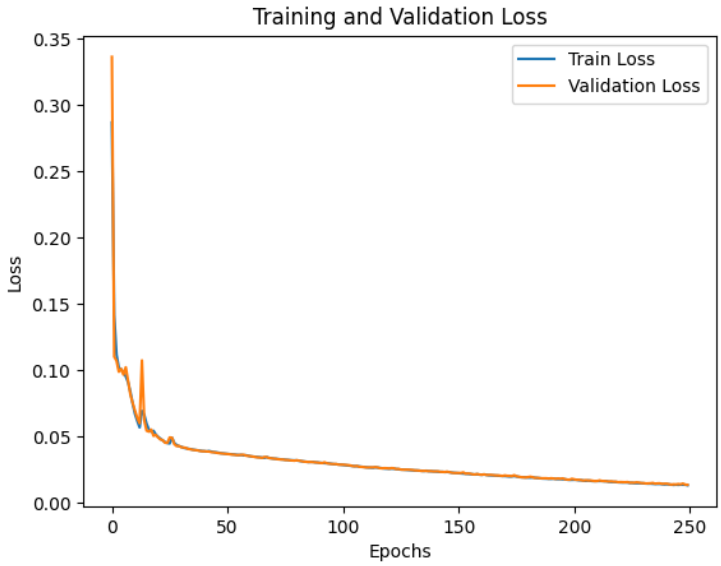

Figure 2. Training and Validation Loss Curve for Autoencoder Model.

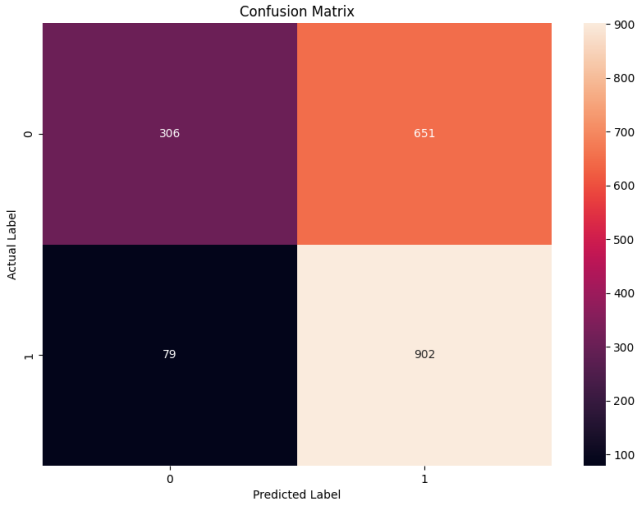

Figure 3. Confusion matrix for Autoencoder Model.

Based on the confusion matrix, the model achieved an accuracy of 62.3%, correctly classifying 1208 out of 1938 total samples. The precision, which measures the proportion of correctly identified malicious samples out of all predicted malicious samples, was 58.1%. The recall, indicating the model's ability to identify all actual malicious samples, was high at 91.9%. In malware detection tasks like this, recall is particularly crucial as failing to identify malicious samples (false negatives) can lead to severe security risks. The F1-score, which balances precision and recall, was 71.2%, reflecting the model's strong performance in detecting malicious samples despite a notable rate of false positives. These results suggest the model is highly sensitive to detecting malware but requires refinement to reduce false positives and improve overall precision.

Conclusion

In this study, we presented an autoencoder-based approach for malware detection, leveraging reconstruction loss to identify anomalies indicative of malicious activity. The results demonstrate the model's ability to achieve high recall (91.9%), indicating its effectiveness in detecting malware, including novel or unknown threats. Study confirms that autoencoders are a promising tool for anomaly-based malware detection, offering a scalable and signature-independent alternative to traditional methods.

To enhance the proposed autoencoder-based approach for malware detection, several directions for future research can be explored. Improving precision is a key priority, as reducing false positives will increase the model’s practicality in real-world applications. This can be achieved by incorporating additional benign data or combining autoencoders with supervised classifiers to create a hybrid detection framework. Expanding the model to include dynamic features, such as API calls, network activity, or runtime behavior, can provide a more comprehensive analysis of malware characteristics.

References

1. Baghirov E. A comprehensive investigation into robust malware detection with explainable AI // *Cyber Security and Applications*. 2025. Vol. 3. Article ID: 100072. ISSN 2772-9184. DOI: 10.1016/j.csa.2024.100072.
2. Baghirov E. Comprehensive framework for malware detection: Leveraging ensemble methods, feature selection, and hyperparameter optimization // *IEEE 17th International Conference on Application of Information and Communication Technologies (AICT)*. Baku, Azerbaijan, 2023. P. 1–5. DOI: 10.1109/AICT59525.2023.10313179.
3. Baghirov E.O. Analyzing the performance of behavioral-based malware detection approaches under real-world conditions // *Optical-Electronic Devices and Devices in Pattern Recognition and Image Processing Systems: Collection of Materials of the XVII International Scientific and Technical Conference*. Kursk, Russia, September 12–15, 2023. P. 15–17. Recognition - 2023 Kursk. South-West State University.
4. Xing, Xiaofei & Jin, Xiang & Elahi, Haroon & Jiang, Hai & Wang, Guojun. A Malware Detection Approach Using Autoencoder in Deep Learning. IEEE Access. 2022. Vol. 10. P. 25696-25706. DOI: 10.1109/ACCESS.2022.3155695.
5. Lee J., Lee J. A classification system for visualized malware based on multiple autoencoder models // *IEEE Access*. 2021. Vol. 9. P. 144786–144795. DOI: 10.1109/ACCESS.2021.3122083.

6. Lakshmanan Nataraj, S Karthikeyan, Gregoire Jacob, and BS Manjunath. Malware images: visualization and automatic classification. In Proceedings of the 8th international symposium on visualization for cyber security. ACM, 4, 2011.
7. Panchagnula V.M., N.V.L. Satya Keerthi Ch., Surekha S., Sujatha R., Veeraiah D., Ramesh E., Lakshmi B. A deep learning approach for detecting malware using autoencoder // *IAENG International Journal of Computer Science*. 2024. Vol. 51, No. 8. P. 1051–1059.
8. Halim M.A., Abdullah A., Zainol Ariffin K.A. Recurrent neural network for malware detection // *International Journal of Advances in Soft Computing and its Applications*. 2019. Vol. 11, No. 1. P. 46–63.
9. Maniriho P., Mahmood A.N., Chowdhury M.J.M. MeMalDet: A memory analysis-based malware detection framework using deep autoencoders and stacked ensemble under temporal evaluations // *Computers & Security*. 2024. Vol. 142. DOI: 10.1016/j.cose.2024.103864.
10. Jin X., Xing X., Elahi H., Wang G., Jiang H. A malware detection approach using malware images and autoencoders // *IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. Delhi, India, 2020. P. 1–6. DOI: 10.1109/MASS50613.2020.00009.
11. Panchagnula V.M., Ch. Satya Keerthi N.V.L., Surekha S., Sujatha R., Veeraiah D., Ramesh E., Lakshmi B. A deep learning approach for detecting malware using autoencoder // *IAENG International Journal of Computer Science*. 2024. Vol. 51, No. 8. P. 1051–1059.
12. Cassavia, N., Caviglione, L., Guarascio, M. *et al.* Learning autoencoder ensembles for detecting malware hidden communications in IoT ecosystems. *J Intell Inf Syst*, 2024. Vol. 62. p. 925–949. https://doi.org/10.1007/s10844-023-00819-8.