Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.061267

ARTICLE

GENOME: Genetic Encoding for Novel Optimization of Malware Detection and Classification in Edge Computing

Sang-Hoon Choi[1] **and Ki-Woong Park**[2,*]

[1]SysCore Lab, Sejong University, Seoul, 05006, Republic of Korea
[2]Department of Computer and Information Security, Sejong University, Seoul, 05006, Republic of Korea
*Corresponding Author: Ki-Woong Park. Email: woongbak@sejong.ac.kr

Received: 20 November 2024; Accepted: 26 January 2025; Published: 06 March 2025

ABSTRACT: The proliferation of Internet of Things (IoT) devices has established edge computing as a critical paradigm for real-time data analysis and low-latency processing. Nevertheless, the distributed nature of edge computing presents substantial security challenges, rendering it a prominent target for sophisticated malware attacks. Existing signature-based and behavior-based detection methods are ineffective against the swiftly evolving nature of malware threats and are constrained by the availability of resources. This paper suggests the Genetic Encoding for Novel Optimization of Malware Evaluation (GENOME) framework, a novel solution that is intended to improve the performance of malware detection and classification in peripheral computing environments. GENOME optimizes data storage and computational efficiency by converting malware artifacts into compact, structured sequences through a Deoxyribonucleic Acid (DNA) encoding mechanism. The framework employs two DNA encoding algorithms, standard and compressed, which substantially reduce data size while preserving high detection accuracy. The Edge-IIoTset dataset was used to conduct experiments that showed that GENOME was able to achieve high classification performance using models such as Random Forest and Logistic Regression, resulting in a reduction of data size by up to 42%. Further evaluations with the CIC-IoT-23 dataset and Deep Learning models confirmed GENOME's scalability and adaptability across diverse datasets and algorithms. The potential of GENOME to address critical challenges, such as the rapid mutation of malware, real-time processing demands, and resource limitations, is emphasized in this study. GENOME offers comprehensive protection for peripheral computing environments by offering a security solution that is both efficient and scalable.

KEYWORDS: Edge computing; IoT security; malware; machine learning; malware classification; malware detection

1 Introduction

Edge computing has emerged as a critical paradigm to meet the increasing demands for real-time analytics and low-latency processing as the technological landscape is reshaped by the rapid proliferation of IoT devices [1–3]. Edge computing, in contrast to centralized cloud computing, reduces the time required for data transmission and improves system responsiveness by bringing computation closer to data sources. Nevertheless, this decentralized approach also introduces substantial security challenges [4–6]. Edge environments are an appealing target for cyberattacks due to their resource-constrained and distributed nature, with malware being one of the most significant threats. Attackers are progressively exploiting vulnerabilities in periphery computing infrastructures to launch sophisticated malware attacks, as evidenced by recent incidents. These environments are subject to a proliferation of malware that is specifically designed to exploit their distinctive architecture, ranging from industrial peripheral systems to IoT networks. For example, the Mirai botnet attack, which compromised IoT devices, underscored the vulnerability of peripheral systems

to large-scale coordinated attacks. At the same time, the necessity of improved security measures in these environments has been emphasized by ransomware campaigns that target peripheral storage systems [7,8].

Extensive research has been conducted to develop effective malware detection and classification techniques in order to address these challenges. Machine Learning (ML) methods have attracted substantial attention due to their capacity to accurately identify and classify malware. Nevertheless, the implementation of ML detection systems in peripheral computing environments is not an easy task. The deployment of resource-intensive ML models and the management of large datasets are significantly impeded by the limited computational resources, memory, and energy constraints of periphery devices. Recent studies have further highlighted the potential of machine learning-based methods for improving malware classification in IoT and edge computing environments [9–13]. These works introduce approaches that balance classification accuracy and computational efficiency, addressing the limitations of resource-constrained systems. Additionally, traditional detection methods frequently fail to maintain pace with the rapid and frequent mutations that malware undergoes as it continues to evolve. Signature-based and behavior-based techniques are the primary methods employed by conventional malware detection systems. Signature-based detection methods are effective in identifying known malware by matching patterns within code. However, they become ineffective when faced with even minor modifications in malware structure. In contrast, behavior-based detection methods are designed to identify malicious activity patterns; however, they are frequently resource-intensive and susceptible to evasion techniques, in which malware imitates benign behavior. These constraints underscore the necessity of a malware detection and classification strategy that is adaptable, robust, and lightweight in peripheral computing environments.

This paper introduces GENOME, a novel framework that is specifically designed to improve the detection and classification of malware in resource-constrained peripheral environments. By extracting artifact information from malware and converting it into DNA-like sequences, GENOME capitalizes on biological inspiration. This innovative representation facilitates the efficient analysis and classification of similarity by encoding the structural and behavioral characteristics of malware into compact, biologically analogous data forms. The detection of evolving malware families is facilitated with greater accuracy by GENOME, which conceptualizes malware evolution to be analogous to viral mutations. A key aim of this research is to demonstrate that even with dataset encoding and compression, the detection and classification performance of GENOME remains robust and consistent. The proposed approach's practicality and effectiveness are validated by our experimental evaluations. The results indicate that GENOME achieves a substantial 42% reduction in data size while maintaining, and in cases, enhancing, classification accuracy compared to conventional methods. These results underscore the viability of implementing GENOME in peripheral computing environments, where detection performance and resource efficiency are of the utmost importance.

This research is a continuation of our previous research, which was presented at the International Conference on Intelligent Information Technology [14]. In contrast to the previous research, which focused on a proof of concept for the concept, this paper investigates the challenges associated with the real-world application and implementation of the approach. The proposed methodology has been validated and proved through a series of extensive experiments.

The remainder of this paper is organized as follows: Chapter 2 reviews related research, Chapter 3 explains the structure and implementation of the GENOME framework, and Chapter 4 presents the evaluation results. Finally, Chapter 5 concludes with discussions on future research directions.

2 Related Work

This section reviews research on malicious behavior detection and classification conducted to enhance the security of edge computing environments.

2.1 Studies on Malware Behavior Detection

Radhakrishna et al. [15] proposed a network edge-based framework for detecting ransomware, demonstrating superior performance compared to existing methods. By employing Random Forest and XGBoost models alongside the chi-square feature selection technique, their model achieved significant improvements. The CICandMal2017 dataset and data augmentation techniques were used to enhance the training process. Ahmed et al. [16] introduced a framework for early ransomware detection in industrial IoT environments. By collecting system call data in a virtual sandbox and applying six machine learning classifiers, they achieved an accuracy of 98.64% with a low false positive rate of 1.7%. Akhil et al. [17] developed a model effective for detecting and classifying malware using a Deep Neural Network (DNN) architecture. DenseNet201 achieved a high accuracy of 94.5%, while MobileNet Small presented a lightweight architecture suitable for real-time malware detection, offering reduced computational latency. Wang et al. [18] achieved an accuracy of 97.55% with a traffic classification model for industrial IoT (IIoT) environments using a semi-supervised learning method. Their approach also optimized latency and improved data privacy. Chuang et al. [19] proposed a malware detection model based on function call graphs and API interaction analysis, reporting an accuracy of 94.5%. However, the study lacked discussion regarding the implementation challenges of the proposed model. Oyler-Castrillo et al. [20] employed Recurrent Neural Networks (RNNs) for malware detection, achieving an accuracy of 98%. However, the study provided no details about training times or real-world deployment feasibility. Almomani et al. [21] achieved an accuracy of 97.5% in detecting Android ransomware using SVM algorithms combined with oversampling techniques. Their model utilized Application Programming Interface (API) and permission characteristics for classification. Taylor et al. [22] reported a detection accuracy of 99.94% by analyzing 19,612 samples of malware and normal data using deep learning.

2.2 Studies on Malware Classification

Akhil et al. [23] emphasized the importance of resource-efficient DNN models for edge computing environments. They suggested that real-time detection requires further optimization of model complexity and false positive rates to address the challenges of evolving malware threats. Aslan et al. [24] reported an accuracy of 97.98% on the Malimg dataset by converting PE files into grayscale images and employing a hybrid network structure. Bae et al. [25] developed a framework leveraging Windows system call sequences, achieving an accuracy of 98.65% through machine learning-based multi-class classification. Yan et al. [26] transformed opcode sequences into grayscale images and attained an accuracy of 99.88% using Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) networks. Singh et al. [27] analyzed Android malware images, achieving a classification accuracy of 92.59% by applying multiple algorithms, including K-Nearest-Neighbor (KNN), Support Vector Machine (SVM), and Random Forest.

As a result, existing studies have shown significant results in detecting and classifying malicious activities, but there are some limitations. First, some studies evaluated models by relying on specific datasets, which may not sufficiently reflect various real-world scenarios. Second, considering the limited resources and real-time processing requirements of edge devices, further optimization of model complexity and latency is required. We applied a DNA-based data representation method that can be efficiently utilized in an edge computing environment through preprocessing of the dataset used in training and designed it to maintain high detection accuracy while reducing the data size.

3 Malware DNA Feature Genetic Encoding Framework

This section introduces the GENOME framework, designed for efficient classification and detection of malicious activities in edge computing environments.

3.1 GENOME Framework

In this study, we propose the GENOME Framework for efficient malicious behavior detection and malware analysis in edge computing environments. This framework systematically collects malicious activity data and converts it into DNA-like sequences for use in training machine learning-based detection models. Fig. 1 shows the structure and main processes of the GENOME Framework, visually explaining the data flow and the role of each component. The GENOME Framework collects malicious activity data in a container-based environment, transfers it to an Analysis Machine, and converts it into DNA sequences using the DNA Generator. This process includes both static data (e.g., binary sections) and dynamic data (e.g., network traffic, system calls, and file system events), effectively extracting and compressing key features associated with malicious activities.

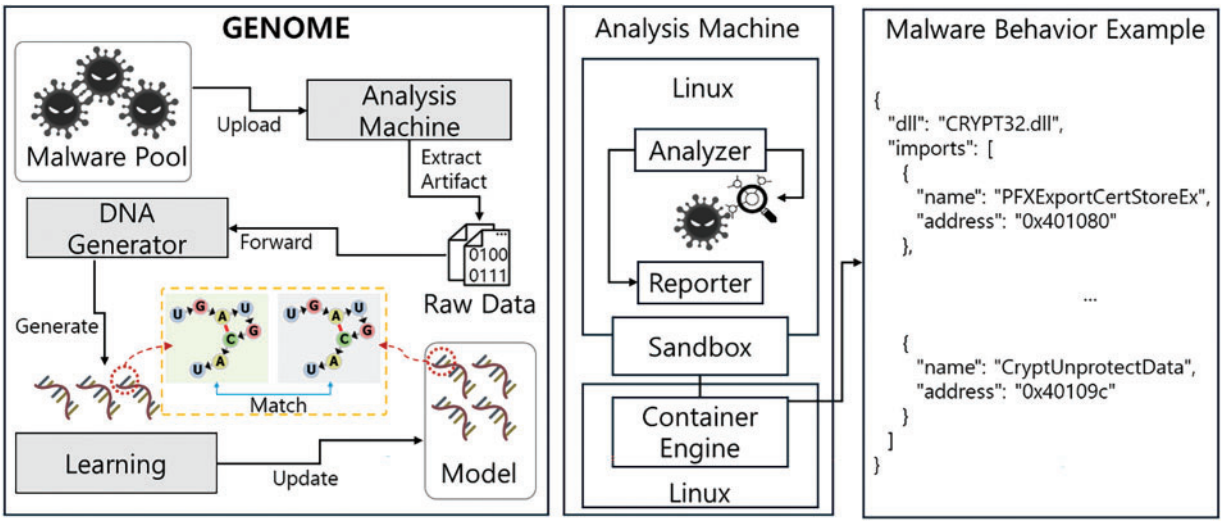

Figure 1: Overview of the GENOME framework

The DNA conversion process involves the following steps: Identifying features associated with malicious activity through binary static analysis. Deriving additional features by collecting dynamic data such as network traffic and system calls. Converting the data into DNA format to reduce dataset size and maximize computational efficiency. The converted DNA data reduces network bandwidth requirements, enhances model training speed, and is optimized for use in resource-limited edge computing environments.

Fig. 1 shows the workflow: malware from the Malware Pool is analyzed and converted into artifact data by the Analysis Machine. The DNA Generator then processes this data into DNA sequences, which are stored as training datasets for machine learning-based detection models. This approach improves the accuracy of malicious activity detection and classification.

Fig. 2 shows the generation process of the DNA Generator. By default, the DNA Generator assigns an artifact 8 bits as a single structure using the sequence table. If DNA compression is enabled, 16 bits are converted into a single DNA structure. The DNA data generated through this process is used to improve the performance of machine learning-based detection models. By considering resource constraints in edge

computing environments, the GENOME Framework compresses data while preserving critical information. This minimizes network traffic, reduces computational load, and enables real-time model training for rapid detection and response to malicious activity, even on edge devices.

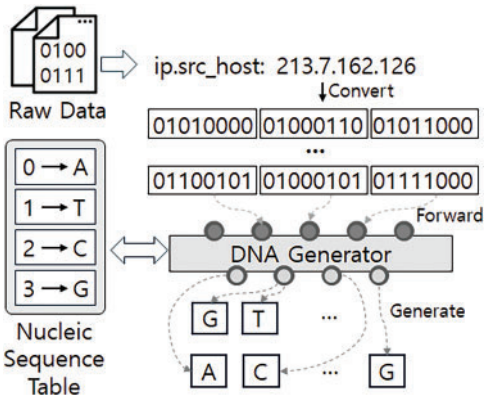

Figure 2: DNA encoding process for network artifacts

3.2 DNA-Based Feature Encoding Algorithms for Malware and Malicious Activities Artifacts

In this study, we propose two DNA encoding algorithms for efficient data processing in edge computing environments: basic DNA encoding (Algorithm 1) and compressed DNA encoding (Algorithm 2). These algorithms take different approaches to converting network traffic data into DNA sequences, enabling efficient representation and processing of large datasets by transforming them into biologically inspired sequences. Table 1 provides detailed information on the data transformation process for network artifacts. The "Original" column refers to the unencoded data, while the "DNA" column represents training data processed using Algorithm 1. The "DNA-Compress" column shows examples of data converted using Algorithm 2.

Algorithm 1: Algorithm for ASCII to DNA encoding

Require: Input ASCII text
Ensure: DNA sequence string
1: **function** ASCII_TO_DNA(ascii_text)
2: initialize empty string dna_sequence
3: initialize DNA_MAP = {0: 'A', 1: 'T', 2: 'C', 3: 'G'}
4: **if** ascii_text = 0 or ascii_text = '0' **then return** '**N**'
5. **end if**
6: **for** each character c in string(ascii_text) **do**
7: ascii_value ← ASCII value of c
8: **if** ascii_value = 0 **then**
9: dna_sequence ← dna_sequence + 'N'
10: **else**
11: remainder ← ascii_value mod 4
12: dna_base ← DNA_MAP[remainder]
13: dna_sequence ← dna_sequence + dna_base

(Continued)

Algorithm 1 (continued)

```
    end if
  end for
  return dna_sequence
end function
```

Algorithm 2: Compressed ASCII to DNA sequence encoding

Require: Input ASCII text
Ensure: Compressed DNA sequence string

```
function COMPRESSED_ASCII_TO_DNA(ascii_text)
    initialize DNA_MAP = {0: 'A', 1: 'T', 2: 'C', 3: 'G'}
    initialize empty string dna_sequence
    if ascii_text is NULL or ascii_text = 0 or ascii_text = '0' then return 'N'
    end if
    ascii_values ← empty array
    for each character c in string(ascii_text) do
        append ASCII value of c to ascii_values
    end for
    for i = 0 to length(ascii_values) step 2 do
        if i + 1 < length(ascii_values) then
            first_value ← ascii_values[i]
            second_value ← ascii_values[i + 1]
            combined_value ← (first_value + second_value) mod 4
        else
            combined_value ← ascii_values[i] mod 4
        end if
        dna_base ← DNA_MAP[combined_value]
        dna_sequence ← dna_sequence + dna_base
    end for
    return dna_sequence
end function
```

Table 1: Example of DNA transformation and compression for network artifacts

Scheme	ip.src_host	ip.dst_host	icmp.checksum
Original	149.40.90.151	192.168.0.128	35,144
DNA	TATCAACTACTTT	TTCCTCACACTCA	GTTAACA
DNA-Compress	TGAGCCT	CAGCCGA	ATCA

Algorithm 1 employs a straightforward approach by mapping each American Standard Code for Information Interchange (ASCII) character to one of four DNA bases: A, T, C, or G. The mapping is based on modulo operations, where the remainder of dividing the ASCII value by 4 determines the base: 0 to A, 1 to T, 2 to C, 3 to G. If the input value is null or zero, an 'N' is assigned to handle exceptions. This one-to-one mapping simplifies data representation while minimizing information loss, preserving the original characteristics of

the data. Algorithm 1's intuitive and clear conversion rules ensure high information retention, making it suitable for applications requiring accurate data restoration.

Algorithm 2 has advanced the basic encoding and added data compression. This algorithm adopted a method of mapping two ASCII characters to one DNA base, which increased the data compression rate. Specifically, the algorithm determines one DNA base by adding the two adjacent ASCII values and using the remainder divided by four. This has the advantage of halving the length of the output data compared to the existing method. Algorithm 2 considers the case where the length of the data sequence is odd, and treats the last single character in the same way as Algorithm 1. This compression method can dramatically reduce storage space while maintaining the overall pattern of the data. In particular, when processing large-scale network traffic data, this compression method allows for efficient use of storage space and transmission bandwidth.

Both methods are engineered for robustness and exception management. They function dependably despite partial or anomalous input data, managing null values and unusual characters proficiently. The time complexity for both methods is O(n), with n representing the length of the input string. Algorithm 1 yields an output length commensurate with the input length, but Algorithm 2 gives output approximately half the length of the input. This compression renders Algorithm 2 more memory-efficient, especially advantageous for handling extensive datasets.

Algorithm 1 ensures total reversibility, facilitating the restoration of the original data. It is appropriate for situations where precision and data integrity are paramount. Conversely, Algorithm 2 is more appropriate for applications that emphasize storage economy and bandwidth conservation, such as edge computing situations with constrained resources. The DNA encoding techniques presented in this paper provide an innovative approach for the analysis and processing of network traffic data. These approaches improve the efficiency of data transmission and processing by converting data into DNA-like sequences. Algorithm 1 is optimal for scenarios necessitating great accuracy, but Algorithm 2 is superior in contexts where the reduction of storage and bandwidth use is critical.

In conclusion, the two suggested algorithms possess unique benefits and limits, permitting their selective use according to the specific environment and requirements. These DNA-based encoding methods provide an innovative method for the effective processing and analysis of network traffic data in edge computing contexts. Subsequent study ought to concentrate on enhancing these encoding techniques and assessing their relevance in other network contexts.

4 Evaluation

This section evaluates the proposed approach by analyzing the GENOME framework's encoding techniques in edge computing environments. The evaluation focuses on classification accuracy and computational efficiency compared to conventional methods.

4.1 Experimental Environment

The experimental environment for this study utilized a system equipped with an Intel Core i7-10700 CPU@2.90 GHz processor, 8 GB of DDR RAM, and a 1 TB SSD for storage. The operating system was Ubuntu 24.04 with kernel version 5.15.0. Malicious activity detection training and performance evaluation were performed in containerized environments, with each container allocated 512 MB of limited memory. Additionally, to verify the scalability of GENOME with Deep Learning models, evaluations were conducted in an NVIDIA RTX 4090 GPU environment.

4.2 Dataset for Performance Evaluation

The Edge-IIoTSet dataset [28] was used to evaluate the performance of the DNA encoding techniques. This dataset is specifically designed for developing intrusion detection systems (IDS) in IoT and Industrial IoT (IIoT) environments and reflects realistic network scenarios. It includes data from various IoT devices, such as temperature, humidity, water level, pH, heart rate, and flame detection sensors. The dataset contains 1,380,858 instances of normal traffic and 546,446 instances of attack traffic, making it suitable for learning complex patterns in IoT/IIoT environments.

Edge-IIoTSet includes 14 attack types, addressing major threats like DoS/DDoS, information gathering, man-in-the-middle attacks, injection attacks, and malware. The dataset provides 61 additional unique features on top of the existing 1176 features to enhance analyzability. It is designed for both centralized machine learning and federated learning environments, offering a high degree of realism compared to other datasets. The added features and large-scale configuration make it an ideal benchmark for evaluating modern machine learning-based intrusion detection models. For additional Deep Learning model evaluations, the CIC-IoT-23 dataset was employed [29]. This dataset incorporates recent IoT attack types and complements the characteristics of the Edge-IIoTSet, enabling a broader assessment.

For these reasons, Edge-IIoTSet was selected as the optimal dataset to assess the performance and practical applicability of the GENOME framework in edge and fog computing environments.

4.3 Performance Comparison of Malware Behavior Detection

The performance of the proposed DNA encoding techniques was evaluated using binary classification models, including SVM and Logistic Regression. The analysis was conducted on three datasets Original, Encoded, and Compressed to examine the effects of data preprocessing and compression on classification performance in edge computing scenarios.

Table 2 summarizes the classification performance of five machine learning models (Random Forest, SVM, K-Nearest Neighbors, Logistic Regression, and Decision Tree) across the three datasets using Precision, Recall, and F1-Score metrics. Random Forest and Decision Tree consistently achieved perfect performance (Precision = 1.0, Recall = 1.0, F1-Score = 1.0) across all datasets. Logistic Regression also maintained perfect performance in both Encoded and Compressed datasets, while SVM and K-Nearest Neighbors experienced slight performance degradation due to the data conversion process.

Table 2: Detection performance of models on original, DNA, and compressed DNA datasets

Dataset	Model	Precision	Recall	F1-Score
Edge-IIoTset (Original)	Random forest	1.00	1.00	1.00
	SVM	1.00	1.00	1.00
	K-nearest neighbors	1.00	0.99	0.99
	Logistic regression	1.00	0.99	0.99
	Decision tree	1.00	1.00	1.00
Edge-IIoTset-DNA (Encoded)	Random forest	1.00	1.00	1.00
	SVM	1.00	0.99	0.99
	K-nearest neighbors	1.00	0.99	0.99
	Logistic regression	1.00	1.00	1.00
	Decision tree	1.00	1.00	1.00
	Random forest	1.00	1.00	1.00

(Continued)

Table 2 (continued)

Dataset	Model	Precision	Recall	F1-Score
Edge-IIoTset-DNA-Compress (Compressed)	SVM	1.00	0.99	0.99
	K-nearest neighbors	1.00	0.99	0.99
	Logistic regression	1.00	1.00	1.00
	Decision tree	1.00	1.00	1.00

Fig. 3 shows the classification results for SVM (top) and Logistic Regression (bottom) using confusion matrices. The matrices display True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) values, offering detailed insights into model prediction accuracy. In the Original dataset, both SVM and Logistic Regression achieved perfect performance (TN = 26,575, TP = 4985, FP = 0, FN = 0), indicating a well-defined feature space and effective learning.

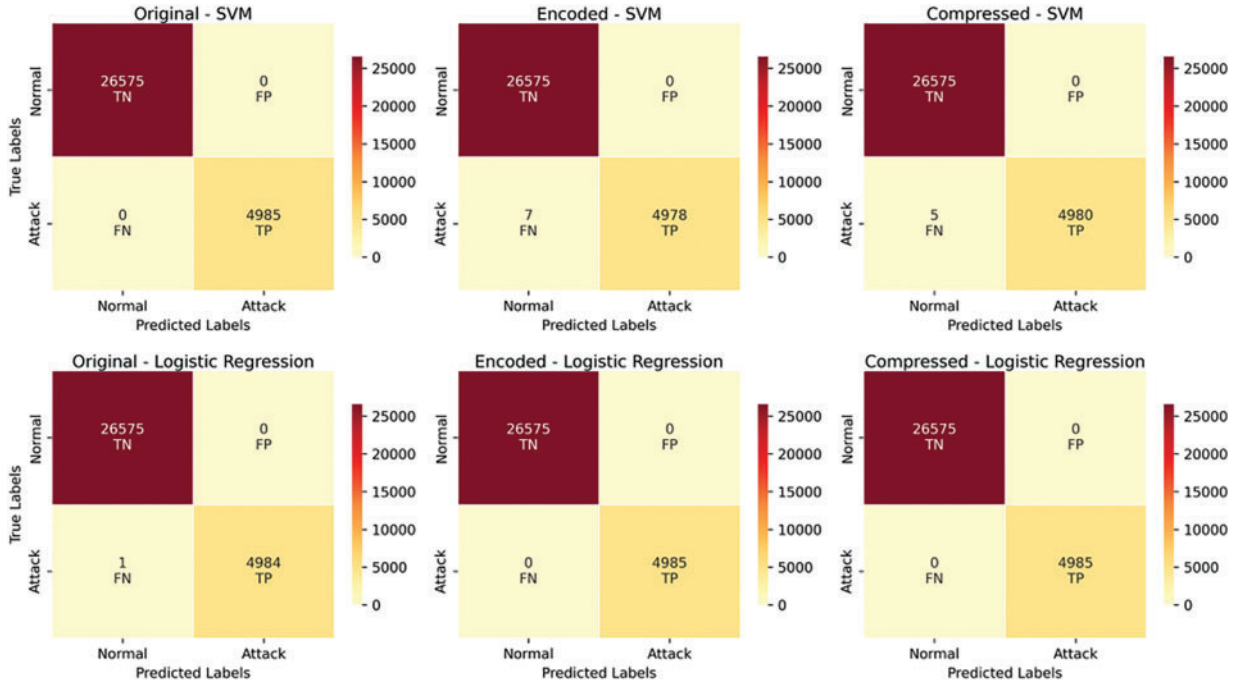

Figure 3: Confusion matrices for SVM and logistic regression

In the Encoded dataset, SVM showed reduced performance with an increase in FN to 7 and a decrease in TP to 4978, suggesting that some attack samples were misclassified as normal. However, Logistic Regression maintained perfect performance (TN = 26,575, TP = 4985, FP = 0, FN = 0), demonstrating its robustness in the Encoded feature space. For the Compressed dataset, SVM slightly improved its performance (FN = 5, TP = 4980) compared to the Encoded dataset but still underperformed relative to the original dataset. Logistic Regression, however, continued to deliver perfect performance (TN = 26,575, TP = 4985), indicating its adaptability to compressed feature spaces. The confusion matrices, with bright colors highlighting TP and TN values, confirm the high accuracy of Logistic Regression and its ability to handle compressed data transformations effectively.

These results reveal that while SVM experienced a slight performance drop in Encoded and Compressed datasets due to margin settings in high-dimensional feature spaces, Logistic Regression maintained consistent accuracy across all datasets. The Encoded and Compressed methods demonstrate significant advantages in edge computing environments by reducing data size and increasing computational efficiency. Although minor performance degradation was observed in specific models, such as SVM, the impact was minimal, and certain models, like Logistic Regression, achieved perfect results. Overall, the DNA conversion techniques proposed in this study offer a practical and efficient solution for data transmission and processing in edge computing environments, balancing resource efficiency with robust performance.

4.4 Performance Comparison of Malware Classification

We analyzed the performance of five machine learning models (Decision Tree, Random Forest, K-Nearest Neighbors, SVM, and Logistic Regression) in terms of Precision, Recall, and F1-Score to evaluate the classification effectiveness of GENOME's data transformation methods. Comparisons were made across three datasets: the original training dataset, the DNA-transformed dataset, and the compressed DNA dataset. This evaluation aimed to identify the correlation between data efficiency and learning performance in edge computing environments.

Tables 3 to 5 present the performance results of each model across the three datasets. In contrast to Table 2, which reflects binary classification, Tables 3 to 5 present multiclass classification results. In multiclass settings, the model must distinguish between multiple attack types, such as Ransomware, Backdoor, and XSS, increasing the complexity of the task. This complexity leads to reduced performance for certain attack types, as observed for Ransomware and Backdoor. The Edge-IIoTset dataset, as the original dataset, was used under optimal training and evaluation conditions for all models. Random Forest and Decision Tree achieved perfect classification performance, consistently maintaining Precision, Recall, and F1-Score values of 1.0 for all attack types, demonstrating their ability to effectively capture complex patterns and nonlinear characteristics. K-Nearest Neighbors exhibited strong performance with F1-Scores exceeding 0.99 in most attack types, though slight performance degradation was observed for Ransomware and Backdoor attacks. SVM showed stable overall performance but recorded marginally lower F1-Scores (0.96–0.97) for XSS and Port Scanning attacks. Logistic Regression demonstrated the weakest performance, particularly for Ransomware (F1-Score = 0.92) and Backdoor (F1-Score = 0.95), reflecting its limitations in learning complex and nonlinear features.

Table 3: Performance of classification models on Edge-IIoTset

Alg	Metr	Normal	Back	Fing	TCP	HTTP	XSS	Scan	SQL	Rans
	Pr	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
DT	Rc	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	f1	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Pr	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99
RF	Rc	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	f1	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Pr	1.00	0.99	1.00	1.00	0.99	0.99	0.99	0.99	0.99
KNN	Rc	1.00	0.99	0.99	0.99	1.00	0.99	0.99	1.00	0.99
	f1	1.00	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
	Pr	1.00	1.00	1.00	1.00	0.99	0.98	0.99	0.99	0.89
SVM	Rc	1.00	0.96	0.91	0.99	1.00	0.99	0.98	1.00	0.99

(Continued)

Table 3 (continued)

Alg	Metr	Normal	Back	Fing	TCP	HTTP	XSS	Scan	SQL	Rans
	f1	1.00	0.98	0.95	0.99	0.99	0.99	0.99	0.99	0.94
	Pr	1.00	0.99	0.81	0.98	0.99	0.98	0.99	0.99	0.91
LR	Rc	1.00	0.95	0.68	0.97	1.00	0.99	0.98	1.00	0.92
	f1	1.00	0.97	0.74	0.98	0.99	0.99	0.99	0.99	0.91

Note: Decision Tree (DT), Random Forest (RF), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Logistic Regression (LR), Algorithm (Alg), Metrics (Metr), Backdoor attack (Back), Fingerprinting attack (Fing), TCP SYN FloodDDoS attack (TCP), HTTP floodDDoS attack (HTTP), Vulnerability scanner attack (Scan), SQL injection attack (SQL), Ransomware attack (Rans), Precision (Pr), Recall (Rc), F1-Score (f1)

Table 4: Performance of classification models on Edge-IIoTset with DNA encoding

Alg	Metr	Normal	Back	Fing	TCP	HTTP	XSS	Scan	SQL	Rans
	Pr	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
DT	Rc	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	f1	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Pr	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99
RF	Rc	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	f1	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Pr	1.00	0.99	1.00	1.00	0.99	0.99	0.99	1.00	0.99
KNN	Rc	1.00	0.99	0.99	1.00	0.99	0.99	0.99	1.00	0.99
	f1	1.00	0.99	0.99	1.00	0.99	0.99	0.99	1.00	0.99
	Pr	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00	0.97
SVM	Rc	1.00	0.97	0.99	1.00	1.00	1.00	0.99	1.00	0.99
	f1	1.00	0.98	0.99	1.00	1.00	1.00	0.99	1.00	0.98
	Pr	1.00	0.93	1.00	1.00	1.00	1.00	1.00	1.00	0.86
LR	Rc	1.00	0.84	0.99	1.00	1.00	1.00	0.99	1.00	0.94
	f1	1.00	0.88	0.99	1.00	1.00	1.00	0.99	1.00	0.90

Table 5: Performance of classification models on edge-IIoTset with compressed DNA encoding

Alg	Metr	Normal	Back	Fing	TCP	HTTP	XSS	Scan	SQL	Rans
	Pr	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
DT	Rc	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	f1	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Pr	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99
RF	Rc	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	f1	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Pr	1.00	0.99	1.00	1.00	0.99	0.99	0.99	1.00	0.99
KNN	Rc	1.00	0.99	0.99	1.00	0.99	0.99	0.99	1.00	0.99
	f1	1.00	0.99	0.99	1.00	0.99	0.99	0.99	1.00	0.99
	Pr	1.00	0.98	1.00	1.00	1.00	0.99	1.00	1.00	0.97
SVM	Rc	1.00	0.97	1.00	1.00	1.00	1.00	0.99	1.00	0.98

(Continued)

Table 5 (continued)

Alg	Metr	Normal	Back	Fing	TCP	HTTP	XSS	Scan	SQL	Rans
	f1	1.00	0.98	1.00	1.00	1.00	0.99	0.99	1.00	0.98
	Pr	1.00	0.70	1.00	1.00	1.00	0.99	1.00	1.00	0.74
LR	Rc	1.00	0.74	1.00	1.00	1.00	1.00	0.99	1.00	0.69
	f1	1.00	0.72	1.00	1.00	1.00	0.99	0.99	1.00	0.72

To explain the high F1-Scores for certain models after data transformation and compression, it is important to note that Random Forest, Decision Tree, and K-Nearest Neighbors adapt well to nonlinear data characteristics. The GENOME framework transforms datasets into a nonlinear feature space through DNA encoding and compression, preserving critical information while removing redundant or noisy details. This transformation highlights essential patterns, enabling these models to effectively learn decision boundaries. In contrast, linear models such as Logistic Regression rely heavily on linear separability and face challenges with nonlinear transformations, leading to performance degradation, particularly for attacks like Ransomware and Backdoor.

The DNA-transformed dataset maintained high classification performance across models. Random Forest and Decision Tree retained perfect scores across all attack types, confirming the transformation process did not impair their learning and inference capabilities. K-Nearest Neighbors consistently achieved F1-Scores above 0.99 for most attack types, with a slight decrease to 0.98 for Ransomware. SVM showed moderate degradation, with F1-Scores of 0.97 for Backdoor and 0.93 for Ransomware, while Logistic Regression exhibited the most significant decline, particularly for Backdoor (F1-Score = 0.88) and Ransomware (F1-Score = 0.90).

The DNA transformation and compression processes retain the variance and structural patterns of the dataset by emphasizing nonlinear characteristics. This aligns with the principles of nonlinear decision boundary learning in models like Random Forest and Decision Tree, which excel at capturing complex interrelationships within the data. These characteristics explain the sustained high classification performance in compressed datasets, particularly for models designed for nonlinear feature spaces. The DNA-Compressed dataset further reduced data size through additional compression, offering advantages in network bandwidth and storage efficiency. Random Forest and Decision Tree continued to deliver perfect performance (Precision, Recall, and F1-Score = 1.0) for all attack types, unaffected by compression. This robustness is attributed to their ability to capture complex patterns and adapt to changes in feature space through ensemble learning and recursive partitioning.

SVM and KNN also performed stably, with high F1-Scores ranging from 0.97 to 0.99 across most attack types. Logistic Regression, however, experienced a significant decline in performance, particularly for Backdoor (F1-Score = 0.74) and Ransomware (F1-Score = 0.72), due to its reliance on linear separability. This highlights the need for nonlinear modeling approaches in scenarios involving transformed and compressed data.

4.5 Evaluation of Training Data Compression

The Edge-IIoTset-DNA and Edge-IIoTset-DNA-Compressed datasets proposed in this study effectively reduce the size of the original dataset (Edge-IIoTset) while maintaining classification performance. Fig. 4 shows the changes in dataset size during the conversion process and evaluates the efficiency of the proposed methods by comparing the Original, Encoded, and Compressed datasets.

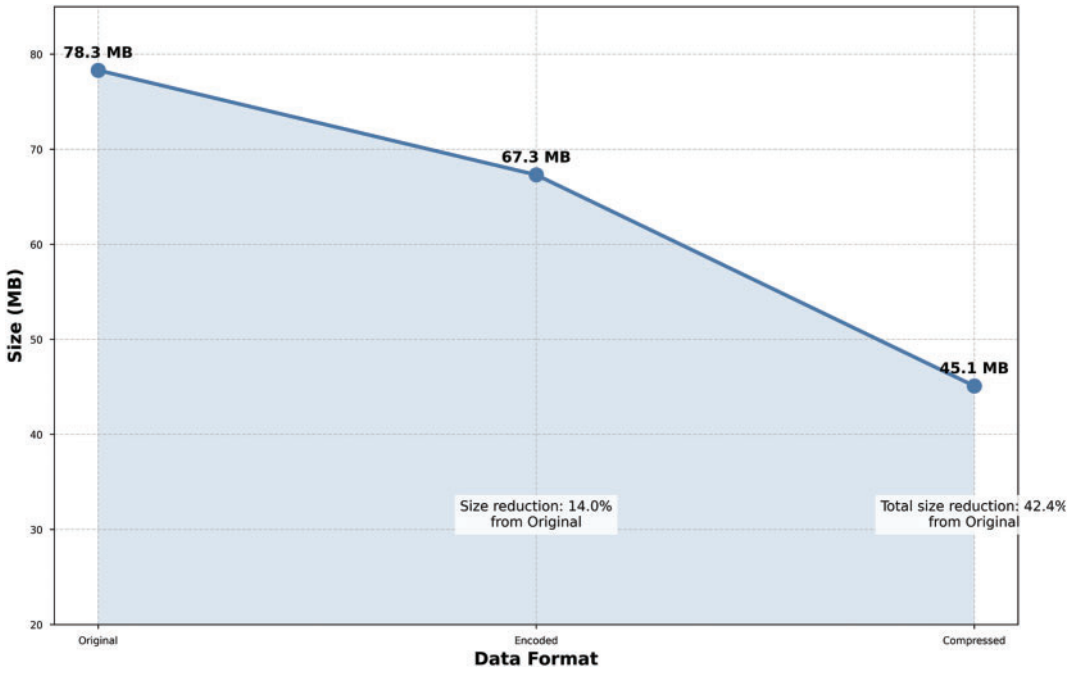

Figure 4: Dataset size reduction across original, DNA-encoded, and DNA-compressed formats

The original dataset, the unprocessed baseline, has a size of approximately 78.3 MB. The Encoded dataset, which applies the DNA encoding method, is reduced to 67.3 MB, achieving a size reduction of about 14%. This demonstrates that DNA-based encoding compresses the data while minimizing information loss. Further applying the DNA-Compressed method results in a dataset size of 45.1 MB, representing a 42.4% reduction compared to the original dataset and an additional 32.9% compression relative to the Encoded dataset. These results highlight the significant potential for reducing data transmission and storage costs in edge computing environments. This reduction in dataset size directly addresses the constraints of edge computing environments, where devices often operate under limited network bandwidth and storage capacity. Large-scale data transfers can lead to bottlenecks, particularly during data synchronization between edge devices and cloud systems. The ability of GENOME to reduce data size without compromising classification performance underscores its transformative impact on resource-constrained IoT systems, enabling practical and scalable solutions for real-time applications. By substantially reducing the size of the dataset, both the Encoded and Compressed formats improve the efficiency of data processing and transmission. The DNA-Compressed dataset, in particular, minimizes transmission costs and ensures that real-time processing requirements are met, making it a critical solution for resource-constrained systems.

In conclusion, as illustrated in Fig. 4, proposed data conversion methods effectively reduce data size while preserving critical information. These findings demonstrate the practicality of the GENOME framework in improving data transmission and storage efficiency, facilitating the deployment and operation of machine learning-based systems in edge computing environments.

4.6 Evaluation of DNA-Based Training Time Performance

KNN consistently required minimal training time approximately 0.016 s across all datasets. This reflects the fact that KNN does not involve a traditional training phase but rather stores data points and performs computations during inference, making it unaffected by dataset size or transformations. Logistic Regression

demonstrated minimal variability in training time, taking 61.00 s for the original dataset, 61.63 s for the Compressed dataset, and 62.57 s for the Encoded dataset. These results suggest that as a simple linear model, Logistic Regression is not significantly impacted by dataset transformations. However, the additional complexity of features in the compressed dataset may have slightly increased computational costs during training. The Decision Tree model showed the most significant improvement, with training time reduced from 1.20 s on the original dataset to 0.66 s on the Compressed dataset, a 45.0% decrease. This highlights the ability of Decision Tree models to efficiently learn from concise representations of data, benefiting from the compressed dataset's reduced size while preserving critical information.

The results indicate that both encoded and compressed datasets significantly decrease training time for the majority of machine learning models. The GENOME framework's ability to accelerate training while maintaining high detection accuracy demonstrates its utility in edge environments, where real-time performance is crucial. The Compressed dataset, on average, attained the briefest training durations, highlighting the efficacy of data transformation techniques in minimizing data volume while preserving critical information for training. Encoded and compressed datasets provide substantial benefits in edge computing settings characterized by restricted network capacity and compute capabilities. They diminish data transmission expenses, enhance data processing efficacy, and facilitate real-time learning and inference. By offering a 42% reduction in dataset size and up to 45% faster training times without compromising detection accuracy, GENOME addresses a critical need in IoT systems for lightweight and efficient solutions that meet real-time demands. The suggested transformation strategy resulted in training time reductions of up to 45.0% relative to the original dataset, while preserving high classification performance. This renders it an effective alternative for reconciling efficiency and accuracy in resource-limited contexts.

In summary, encoded and compressed datasets significantly decrease training duration and enhance resource efficiency, rendering them suitable for edge computing contexts. Fig. 5 demonstrates that the observed enhancements in training duration highlight the practicality and applicability of this method for real-world scenarios.

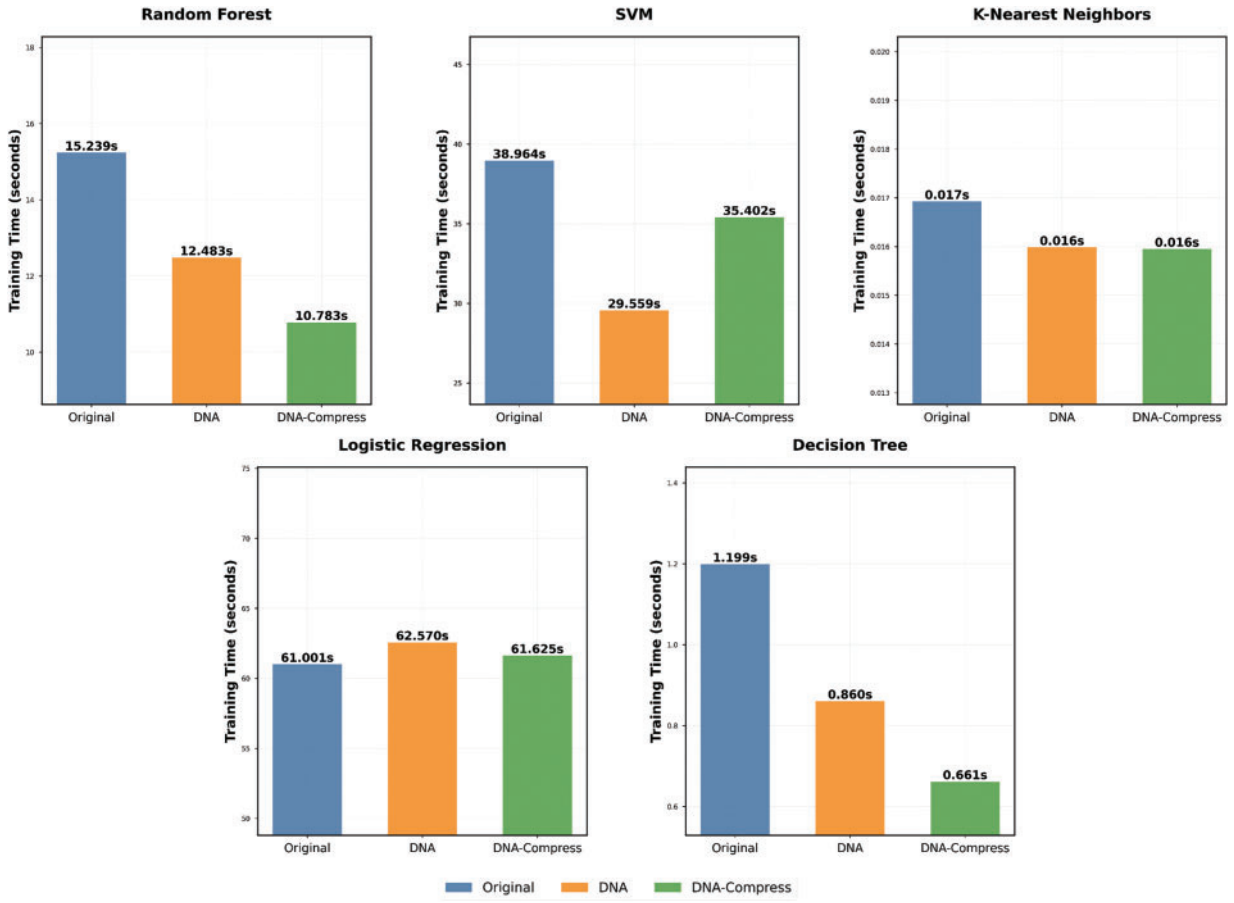

Figure 5: Training time comparison of machine learning models on original, DNA-encoded, and DNA-compressed datasets

4.7 Verification of GENOME Framework Scalability

To verify its applicability in resource-constrained IoT environments, the GENOME framework was initially evaluated using ML models. These evaluations showed that GENOME could reduce dataset size by up to 42% while decreasing training time for certain ML models by as much as 45%. However, the scope of these evaluations was limited to the Edge-IIoTset dataset and ML models, leaving questions about GENOME's broader scalability and versatility unanswered.

To address this gap, additional experiments were carried out to explore how GENOME performs when applied to Deep Learning (DL) models and other datasets. For this purpose, we used the CIC-IoT-23 dataset, which is a modern, real-world dataset that captures diverse IoT network traffic and attack scenarios. Specifically, we focused on classifying three major attack types: DDoS-ICMP (36,554 records), DDoS-PHSACK (21,210 records), and DDoS-SynonymousIP (18,189 records). To ensure consistency across experiments, all models were evaluated using default hyperparameters.

This extended evaluation incorporated three DL models—DNN, CNN, and LSTM—as well as two cutting-edge ML models: XGBoost and LightGBM. XGBoost was chosen for its efficiency in handling large-scale data and its ability to deliver high accuracy in high-dimensional datasets, making it particularly valuable for IoT environments where identifying key features is critical. LightGBM, on the other hand, was selected

for its speed and memory efficiency, which make it highly suitable for resource-limited IoT devices. Its ability to perform real-time analysis in large-scale datasets further supports its use in such scenarios.

Table 6 presents the results of these experiments. Interestingly, GENOME's performance on the CIC-IoT-23 dataset was largely consistent with the earlier results on Edge-IIoTset. While some models exhibited slight performance improvements or declines, these variations could likely be addressed through further optimization of hyperparameters. Overall, the framework demonstrated strong adaptability to both the dataset and the models used. The results underline the robustness of GENOME's data transformation and compression techniques. Nonlinear models such as Random Forest, Decision Tree, and K-Nearest Neighbors continued to perform exceptionally well, taking full advantage of GENOME's ability to retain critical patterns while discarding unnecessary details. Similarly, DNN and CNN demonstrated stable performance, likely due to their inherent compatibility with the nonlinear patterns emphasized by GENOME's encoding. LSTM, however, showed slight declines in performance for certain attack types, potentially because its reliance on temporal features was not fully addressed by the encoding process. These findings confirm that GENOME is not only scalable but also versatile, adapting seamlessly across a range of algorithms and datasets. Its ability to reduce data size without compromising real-time processing capabilities makes it an ideal solution for IoT environments, where resource constraints are a significant challenge. The framework's demonstrated effectiveness with both traditional ML and advanced DL models highlights its potential for widespread adoption in edge computing systems.

Table 6: Evaluation of classification models on IoT-23 dataset using GENOME framework

Alg	Metr	IoT-23			IoT-23 DNA encoding			IoT-23 compressed DNA encoding		
		ICMP	PSHACK	SynIP	ICMP	PSHACK	SynIP	ICMP	PSHACK	SynIP
DNN	Pr	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	1.0
	Rc	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
	f1	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
CNN	Pr	0.99	0.99	1.0	0.99	0.99	0.99	0.99	0.99	1.0
	Rc	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
	f1	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
LSTM	Pr	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	1.0
	Rc	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
	f1	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
XGBoost	Pr	1.0	1.0	1.0	1.0	0.99	0.99	0.99	0.99	0.99
	Rc	1.0	1.0	1.0	1.0	1.0	0.99	0.99	0.99	0.99
	f1	1.0	1.0	1.0	1.0	0.99	0.99	0.99	0.99	0.99
LightGBM	Pr	1.0	1.0	1.0	0.99	1.0	1.0	0.99	0.99	0.99
	Rc	1.0	1.0	1.0	1.0	1.0	0.99	0.99	1.0	0.99
	f1	1.0	1.0	1.0	0.99	1.0	0.99	0.99	0.99	0.99

Note: Algorithm (Alg), Metrics (Metr), Backdoor attack (Back), DDoS-ICMP_Flood (ICMP), DDoS-PSHACK_Flood (PSHACK), DDoS-SynonymousIP_Flood (SynIP), Precision (Pr), Recall (Rc), F1-Score (f1)

In summary, this extended evaluation validates GENOME's scalability and applicability beyond its initial scope. By delivering consistent performance across diverse models and datasets, GENOME provides a practical and efficient framework for addressing the unique challenges of IoT environments.

5 Conclusion

This study proposed an innovative DNA-based framework, GENOME, for malware detection and classification in edge computing environments. GENOME tackles significant issues in edge environments by converting malware artifacts into DNA-like sequences, hence minimizing data size and facilitating effective processing. GENOME attained substantial enhancements in data size reduction and detection accuracy through two methodologies: basic DNA encoding and compressed DNA encoding. Employing the Edge-IIoTset dataset, GENOME achieved a data size reduction of up to 42% while preserving superior classification performance in models including Random Forest and Decision Tree. The results indicate that GENOME is an effective solution for malware detection in resource-limited edge situations. Specifically, compressed DNA encoding markedly diminished network bandwidth consumption and computing demands, rendering it very appropriate for real-time applications. Nonetheless, a decline in performance was noted in linear models, including Logistic Regression, when utilizing compressed datasets. This underscores that, although GENOME adeptly combines data efficiency and detection accuracy, the selection of the machine learning model is essential for achieving optimal outcomes. To further validate GENOME's scalability and adaptability, experiments with Deep Learning models and the CIC-IoT-23 dataset demonstrated its effectiveness across diverse datasets and algorithms, reinforcing its potential for broader applicability in modern IoT scenarios.

In summary, GENOME offers a pragmatic and effective method for malware detection and classification in edge computing settings. By attaining equilibrium between data efficiency and detection efficacy, it tackles the principal security constraints of edge systems and illustrates its capability to resolve these issues in practical applications. Future research will concentrate on the development and assessment of RNA-based algorithms that consider sequence integrity and temporal relationships. We anticipate that this method will enhance the monitoring and identification of evolving malware. By converting malware artifacts into RNA-like sequences, we seek to enhance the identification of both known and modified malware, hence improving the efficacy of malware detection systems in dynamic edge environments.

Acknowledgement: The authors would like to express their gratitude to the editors and anonymous reviewers for their valuable feedback and constructive suggestions.

Funding Statement: This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) (Project Nos. RS-2024-00438551, 30%; 2022-11220701, 30%; 2021-0-01816, 30%), and the National Research Foundation of Korea (NRF) grant funded by the Korean Government (Project No. RS2023-00208460, 10%).

Author Contributions: The authors confirm their contributions to the paper as follows: Sang-Hoon Choi contributed to the study conception and design, methodology, and software development; drafted the manuscript; and conducted validation and formal analysis. Ki-Woong Park contributed to the study conception and design, funding acquisition, and supervision; and participated in drafting, reviewing, and editing the manuscript. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Aloi G, Caliciuri G, Fortino G, Gravina R, Pace P, Russo W, et al. Enabling IoT interoperability through opportunistic smartphone-based mobile gateways. J Netw Comput Appl. 2017;81(7):74–84. doi:10.1016/j.jnca.2016.10.013.
2. Fortino G, Savaglio C, Zhou M. Toward opportunistic services for the industrial Internet of Things. In: 2017 13th IEEE Conference on Automation Science and Engineering (CASE); 2017 Aug 20–23. Xi'an; China: IEEE; 2017. p. 825–30. doi:10.1109/COASE.2017.8256205
3. Wang S, Zhang X, Zhang Y, Wang L, Yang J, Wang W. A survey on mobile edge networks: convergence of computing, caching and communications. IEEE Access. 2017;5:6757–79. doi:10.1109/ACCESS.2017.2685434.
4. Kaur B, Dadkhah S, Shoeleh F, Neto ECP, Xiong P, Iqbal S, et al. Internet of Things (IoT) security dataset evolution: challenges and future directions. Internet Things. 2023;22(10):100780. doi:10.1016/j.iot.2023.100780.
5. Mazhar T, Talpur DB, Al Shloul T, Ghadi YY, Haq I, Ullah I, et al. Analysis of IoT security challenges and its solutions using artificial intelligence. Brain Sci. 2023;13(4):683. doi:10.3390/brainsci13040683.
6. Azrour M, Mabrouki J, Guezzaz A, Kanwal A. Internet of Things security: challenges and key issues. Secur Commun Netw. 2021;2021(3):5533843. doi:10.1155/2021/5533843.
7. Humayun M, Jhanjhi N, Alsayat A, Ponnusamy V. Internet of Things and ransomware: evolution, mitigation and prevention. Egypt Inform J. 2021;22(1):105–17. doi:10.1016/j.eij.2020.05.003.
8. Saeed S, Jhanjhi NZ, Naqvi M, Humayun M, Ahmed S. Ransomware: a framework for security challenges in Internet of Things. In: 2020 2nd International Conference on Computer and Information Sciences (ICCIS); 2020 Oct 13–15; Sakaka, Saudi Arabia: IEEE; 2020. p. 1–6. doi:10.1109/iccis49240.2020.9257660
9. Alwahedi F, Aldhaheri A, Ferrag MA, Battah A, Tihanyi N. Machine learning techniques for IoT security: current research and future vision with generative AI and large language models. Internet Things Cyber Phys Syst. 2024;4(4):167–85. doi:10.1016/j.iotcps.2023.12.003.
10. Farfoura ME, Alkhatib A, Alsekait DM, Alshinwan M, El-Rahman SA, Rosiyadi D, et al. A low complexity ML-based methods for malware classification. Comput Mater Continua. 2024;80(3):4833–57. doi:10.32604/cmc.2024.054849.
11. Maray M, Alqahtani H, Alotaibi SS, Alrayes FS, Alshuqayran N, Alnfiai MM, et al. Optimal bottleneck-driven deep belief network enabled malware classification on IoT-cloud environment. Comput Mater Contin. 2023;74(2):3101–15. doi:10.32604/cmc.2023.032969.
12. Mothukuri V, Khare P, Parizi RM, Pouriyeh S, Dehghantanha A, Srivastava G. Federated-learning-based anomaly detection for IoT security attacks. IEEE Internet Things J. 2022;9(4):2545–54. doi:10.1109/JIOT.2021.3077803.
13. Khan MM, Buriro A, Ahmad T, Ullah S. Backdoor malware detection in industrial IoT using machine learning. Comput Mater Contin. 2024;81(3):4691–705. doi:10.32604/cmc.2024.057648.
14. Choi SH, Park KW. DRACULA: dynamic RNA-based analysis for classifying unseen and lurking attacks. In: Proceedings of the International Conference on Smart Mobility and Revolutionary Transportation; 2024. p. 179–83.
15. Radhakrishna T, Majd NE. Edge computing ransomware detection in IoT using machine learning. In: 2024 International Conference on Computing, Networking and Communications (ICNC); 2024 Feb 19–22; Big Island, HI, USA: IEEE; 2024. p. 244–8. doi:10.1109/ICNC59896.2024.10556351
16. Ahmed YA, Huda S, Ali Saleh Al-rimy B, Alharbi N, Saeed F, Ghaleb FA, et al. A weighted minimum redundancy maximum relevance technique for ransomware early detection in industrial IoT. Sustainability. 2022;14(3):1231. doi:10.3390/su14031231.
17. Akhil MR, Sharma AKV, Swamy H, Pavan A, Shetty A, Sathyanarayana AB. Malware classification using deep neural networks: performance evaluation and applications in edge devices. arXiv:2310.06841. 2023.
18. Wang M, Zhang B, Zang X, Wang K, Ma X. Malicious traffic classification via edge intelligence in IIoT. Mathematics. 2023;11(18):3951. doi:10.3390/math11183951.
19. Chuang HY, Chen JL, Ma YW. Malware detection and classification based on graph convolutional networks and function call graphs. IT Professional. 2023;25(3):43–53. doi:10.1109/MITP.2023.3264509.

20. Oyler-Castrillo M, Agostini NB, Sznaier G, Kaeli D. Machine learning-based malware detection using recurrent neural networks. In: IEEE MIT Undergraduate Research Technology Conference (URTC); 2019 Oct 11–13. Cambridge, MA, USA: IEEE; 2019. p. 1–4. doi:10.1109/urtc49097.2019.9660435
21. Almomani I, Qaddoura R, Habib M, Alsoghyer S, Al Khayer A, Aljarah I, et al. Android ransomware detection based on a hybrid evolutionary approach in the context of highly imbalanced data. IEEE Access. 2021;9:57674–91. doi:10.1109/ACCESS.2021.3071450.
22. Taylor OE, Ezekiel PS, Sako DJS. A deep learning based approach for malware detection and classification. Int J Soft Hard Res Eng (IJSHRE). 2021;9(4):32–40.
23. Akhil MR, Sharma AKV, Nadig A, Pavan A, Shetty A, Sumathi A. Deep learning for cyberthreats: performance analysis and application of malware classification in edge computing. In: 2024 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE); 2024; Bangalore, India; p. 1–5. doi:10.1109/IITCEE59897.2024.10467665
24. Aslan Ö, Yilmaz AA. A new malware classification framework based on deep learning algorithms. IEEE Access. 2021;9:87936–51. doi:10.1109/ACCESS.2021.3089586.
25. Bae SI, Lee GB, Im EG. Ransomware detection using machine learning algorithms. Concurr Computat. 2020;32(18):e5422. doi:10.1002/cpe.5422.
26. Yan J, Qi Y, Rao Q. Detecting malware with an ensemble method based on deep neural network. Secur Commun Netw. 2018;2018(1):7247095. doi:10.1155/2018/7247095.
27. Singh J, Thakur D, Ali F, Gera T, Kwak KS. Deep feature extraction and classification of Android malware images. Sensors. 2020;20(24):7013. doi:10.3390/s20247013.
28. Ferrag MA, Friha O, Hamouda D, Maglaras L, Janicke H. Edge-IIoTset: a new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning. IEEE Access. 2022;10:40281–306. doi:10.1109/ACCESS.2022.3165809.
29. Neto ECP, Dadkhah S, Ferreira R, Zohourian A, Lu R, Ghorbani AA. CICIoT2023: a real-time dataset and benchmark for large-scale attacks in IoT environment. Sensors. 2023;23(13):5941. doi:10.3390/s23135941.