International Journal of Advanced Technology and Engineering Exploration, Vol 12(124)
ISSN (Print): 2394-5443 ISSN (Online): 2394-7454
http://dx.doi.org/10.19101/IJATEE.2024.111101413

Research Article

A comparative study of deep learning-based ransomware detection for industrial IoT

Deo Irankunda*, Khalid El Fazazy, Tairi Hamid and Jamal Riffi
Department of Computer Science, Faculty of Science Dhar El Mahraz, University of Sidi Mohamed Ben Abdellah of Fes, Morocco

Received: 05-August-2024; Revised: 19-February-2025; Accepted: 22-February-2025

Abstract
Currently, internet usage plays a crucial role in industrial development, serving as a source of knowledge and a communication channel. Each year, industries increasingly integrate digital capabilities into their daily operations. The internet of things (IoT) is an emerging technology that offers numerous advantages, including enhanced industrial processes, increased efficiency, and improved visibility. However, it also expands the attack surface for cyber-physical threats. Ransomware attacks are among the most severe malware threats in the industrial internet of things (IIoT), primarily focusing on encrypting files and restricting access to critical industrial systems. Victims often face the obligation of paying a ransom to regain access. Detecting malware and intrusions in IIoT environments requires advanced techniques, including artificial intelligence tools, to identify malicious activities and unauthorized access. This study employs a descriptive comparison method to analyze the structure, advantages, and limitations of deep learning models, including generative adversarial networks (GANs), autoencoders (AE), long short-term memory (LSTM), bidirectional long short-term memory (Bi-LSTM), and convolutional neural networks (CNNs). Additionally, opcode sequences are combined with high-order n-grams to enhance ransomware detection accuracy. This system extracts opcodes from executable files and analyzes their patterns to identify malicious code. Furthermore, a prescriptive analysis of each model's hyperparameters is performed, and their performance is evaluated using ransomware portable executable (PE) header features and the IoT-23 dataset. The TensorFlow framework is utilized to capture temporal dependencies and mitigate vanishing gradient issues. The results demonstrate the superior performance of the LSTM and CNN models, achieving an accuracy of 96.98%, a precision of 97.10%, a recall of 97.00%, and an F1-score of 96.98%.

Keywords
Ransomware detection, Industrial internet of things (IIoT), Deep learning models, Malware analysis, Opcode sequences, Cybersecurity in IIoT.

1.Introduction
The internet of things (IoT) plays a significant role in industrial transformation, giving rise to the term Industrial internet of things (IIoT). This transformative technology interconnects various devices, such as sensors and actuators, enabling data collection and communication over the internet [1]. The IIoT (Industry 4.0) is an emerging technology that revolutionizes manufacturing and production through embedded networking, sensing devices, and actuators integrated with edge computing technology. It is used to enhance production, accuracy, and visibility in various sectors including healthcare, education, automation systems, smart grids, smart cities, the military, and the environment [2].

IIoT merges key infrastructures such as operational technology (OT) systems with information technology (IT). Manufacturing industries are transitioning to smart factories by linking their OT networks to cloud platforms [3]. Advanced threats targeting critical operations, such as production devices, have emerged, increasing the vulnerability of industrial networks to cyberattacks.

Ransomware, known as a virus variant, is a form of malicious software that restricts access and encrypts data until the payment of the ransom to regain access to the computer or data [4]. Once ransomware attacks succeed, several effects include information security breaches, machine downtime, and financial losses [5]. Ransomware attacks often exploit system vulnerabilities that have not been addressed by manufacturers or IT teams. These attacks pose a

*Author for correspondence

significant threat to industries, as they can target multiple entities by restricting access to critical computers or central network devices essential for industrial operations. Ransomware attackers aim to maintain control until a ransom is paid, potentially causing irreversible damage [6]. According to the Cybersecurity Ventures report of July 2023, which analyzed data over six years (2015–2021), and the internation business machines (IBM) Security X-Force report of 2021, ransomware has significantly evolved in terms of the scale of damages. In 2015, ransomware-related damages amounted to $325 million; by 2017, they had surged to $5 billion, marking a 15-fold increase. The estimated damages continued to rise, reaching $8 billion in 2018, $11.5 billion in 2019, and $20 billion in 2021. By 2021, ransomware damages had increased 57-fold compared to 2015. Projections indicated that by 2024 damages will reach $42 billion—an increase of 129.23 times the 2015 amount [7]. The rapid expansion of connected IoT devices is exacerbating cyberattack risks, as attackers increasingly exploit vulnerabilities within these devices.

Information security ensures users' and devices' safety through tools and security measures. Intrusion detection systems (IDSs) are used for network monitoring, malicious activities or unusual activity detection, and alarming systems [8]. Firewall devices are the network gatekeepers and can monitor communications and examine and filter network traffic, allowing only permitted access [9]. However, these tools have some limitations and weaknesses involving the reinforcement of their capacities to build efficient security systems. Artificial intelligence and its branches are advanced technological tools with advantages in multiple application domains, including cybersecurity. They have been used to detect and classify malware attacks by analyzing their behavior and extracting features that allow them to differentiate malware from their families [10]. Numerous research studies have explored and utilized the advantages of deep learning models, a subset of machine learning (ML) that employs multi-layered neural networks to learn complex patterns from data. These models serve as powerful tools, enabling security professionals to develop hybrid systems that enhance information security, enable real-time threat detection and alerts, and provide solutions to complex cybersecurity challenges [11]. This study examines the structure, benefits, and limitations of generative adversarial networks (GANs), autoencoders (AE), long short-term memory (LSTM), bidirectional long short terms memory (Bi-LSTM), and convolutional neuron networks (CNNs) deep learning model-based ransomware detection in IIoT and highlights recommended practices after evaluating their performance.

This study contributes to real-time ransomware detection by employing opcode analysis and high-order n-grams to understand the behavior of executable files. Additionally, the TensorFlow framework is utilized to capture temporal dependencies and mitigate vanishing gradient challenges. This approach effectively enhances the detection of new ransomware variants without relying on pre-labeled data while ensuring high efficiency in pattern recognition to combat ransomware attacks.

The structure of this paper is as follows: Section 2 presents the literature review and related work. Section 3 outlines the materials, methods, and approaches used, including the architectures, advantages, limitations of each model, and experimentation parameters. Section 4 presents the results, while Section 5 discusses the findings. Finally, Section 6 concludes the study.

2.Literature review

2.1Industrial IoT security challenges

The IIoT is an emerging technology that expands significantly and creates more collaboration based on its infrastructure. Its infrastructures comprise many heterogeneous devices connected through communication networks [12]. Ensuring data confidentiality, availability, and integrity is challenging, resulting in mistrust of network operations and potentially losing sensitive information. A botnet is a set of compromised computers that are remotely controlled by a botmaster to spread malware, steal data, and launch distributed denial of services (DDoS) [13]. Another challenge for IIoT infrastructure is legacy OT and software. As organizations strive to meet the increasing demands each year, they often produce machines, actuators, and embedded sensors without implementing adequate security measures, such as patch management options [14]. Most of these machines have outdated software and hardware, with vulnerabilities that can pause production cycles and cause downtime. Some cyberattacks, such as the WannaCry ransomware attack, can happen through this vulnerability and remain undetected [15]. Implementing severe measures for forensics, upgrading, and access control to reinforce security is required [16]. Thus, some of the used communication

protocols, such as Modbus and Profinet, present vulnerabilities that enable attackers to get access by deploying tactics like self-propagation worms and peer-to-peer C2 communications for disrupting critical processes [17]. IoT and IT devices also face interoperability challenges. Most industrial ecosystems have multiple interconnected devices and software from different vendors, which becomes a root configuration error and opens the door to attackers [18]. Finally, the IoT domain still faces unclear regulatory guidelines. There are currently no official regulations in place to define how IoT data will be collected, accessed, and shared, potentially affecting privacy [19].

2.2Intrusion detection in IoT

Detection is identifying unusual activity and alerting users before explosions. It enables us to find infection earlier or unusual activities and take precautions and measures. Traditionally, ransomware and other types of malware detections have been made using various techniques, such as behavior-based, signature-based, and deception-based [20] presented in intrusion detection tools. This has significantly contributed to cybercriminal attack protection [21]. *Table 1* compares IDS systems by showcasing the deployed techniques, advantages, and limitations of anomaly detection.

Table 1 Intrusion detection in the IoT

Tools	Categories	Advantages	Limitations
Traditional [20]	Signature-based	Effective for known signature, low false positive, fast response.	Ineffective for new variant attacks, it struggles with encrypted files,
	Behavior-based	Excellent for stopping hackers and malicious programs from exfiltrating data and identifying all strange behaviors in the system.	Time-consuming, ineffective for legitimate file sharing sites or IP add attacks, resource consuming, high false positive.
	Deception based	Effective for new variant detection.	It is time-consuming to set up and maintain. It has high false positives and is a challenging honeypot.
	Hybrid	Increased accuracy, and greater flexibility.	High complexity, costly
IDS	Cloud-based	Centralized management, scalability, low power consuming.	Privacy concerns, high bandwidth, and latency.
	Edge-based	Low latency and increased data privacy.	Resources constraints
	Hybrid	Increased flexibility and accuracy.	Very complex.

Currently, artificial intelligence and its subsets are good tools used to overcome the limitations of these techniques, as mentioned in *Table 1*[22].

2.3Related works

Each year, numerous industries integrate IoT technology and processes to enhance operational efficiency. However, as IoT networks drive significant industrial advancements, they also become attractive targets for cybercriminals. Extensive research has been conducted to address attacks targeting IIoT, leading to various proposed solutions. *Table 2* summarizes the authors' contributions to this field and the drawbacks of these existing works.

Table 2 Existing works

Research paper	Paper contribution	Mythology and algorithms used	Dataset used	Results	Drawbacks
[23], 2022	Development of a new strategy to overcome heterogeneous ransomware attacks by extracting multiple core features.	Cost-Sensitive Pareto Ensemble framework that is based on dynamic analysis and the study used random forest (RF) and computer science lab	-582 ransomware samples -942 benign -11 families	Effective improvement of performance against zero-day ransomware attacks. Recall 9%	The model used an old dataset, and it is time-consuming.

Research paper	Paper contribution	Mythology and algorithms used	Dataset used	Results	Drawbacks
		(LRC) algorithms.			
[24], 2020	A deoxyribonucleic acid (DNA) sequencing engine for ransomware detection and classification. was developed.	DNAct-Ran method based on genome rules for ransomware detection. The study used the RF algorithm.	-582 ransomware samples -942 benign	Accuracy: 78.5% Precision:75.8% Recall:83.3% F1-sore 87.9%	The proposed model has low performance.
[25], 2022	Single and combined CNN and LSTM models were developed for malware detection and classification in IoT devices.	This study used a comparative analysis to measure the efficiency of single and hybrid models. The study deployed LSTM and CNN independently and then both together. Finally, the hybrid model provided better accuracy than the single models.	-Malware samples: 128 -Benign samples: 1089.	Accuracy:99.83% Precision:99.80% Recall:99.79% F1-sore:99.71%	The model performs well in detecting malware in home appliances but struggles in large IoT networks.
[26], 2023	A boosted CNN and ensemble learning model was developed to detect malware in IoT networks. The proposed model enhanced the learning rate and provided a robust real-time framework for analyzing and detecting malicious activity.	DSBEL framework comprises blocks and global boundaries able to learn from features to identify malware patterns IoT networks have been used in this study. To evaluate the model's performance, the study implemented a CNN deep learning model.	-1473ransomware samples -2486 benign samples	Accuracy:98.50% Precision:98.42% Recall:95.97% F1-sore:97.12%	The model is time-consuming and has a high false positive rate (FPR).
[27], 2022	This study proposed a machine-learning model to analyze and distinguish malware and benign in less time. The researcher used different algorithms to find the best one that could perform detection with better accuracy.	This study used a comparative analysis method.to find the best algorithm for malware detection among RF, support vector machine (SVM), and Naïve Bayes (NB).	-APK samples of 631956 -80% of samples have been used for training -20% for testing.	Accuracy: 99.47% FPR:1.70%	The model is not efficient for the dataset of less adware samples.
[28], 2023	This study aimed to implement multiple ML classifiers and compare the gotten results to identify the suitable ML approach for intrusion detection in IoT networks.	This study deployed machine ML classifier techniques based on communication protocols, such as Message Queuing Telemetry Transport (MQTT). It also used various algorithms, such as decision tree (DT), k-nearest neighbor (K-NN), RF, SVM, and NB.	MQTT-IoT-IDS dataset	All used algorithms proved their high performance with below accuracy: RF:99.98%, DT:99.98%, K-NN:97.76%, SVM:97.80%, NB: 97.58%	This study used old datasets which are not effective in detecting new ransomware variants
[29], 2022	The study developed a new deep-learning approach to detect malicious-based image representations of binary files. It also contributes to classifying unknown variants of ransomware.	This study used Malware SMELL-based new shot learning to classify ransomware using visual representation. It remedied classification challenges by enhancing class separability.	This study used a mailing dataset containing 9339 samples from 25 ransomware families. This dataset doesn't contain benign codes.	Accuracy:84.01% Precision:85.06% Recall:80.00% F1-sore:81.02%	The performance of the proposed study is low. Some attacks could remain undetected.
[30], 2023	An optimal graph of CNN-based Ransomware detection in an IoT	This study used the CNN model with an optimal graph	The evaluation was conducted using a dataset of 420	The model performed well as shown in the following metrics	The proposed model is time-consuming.

Research paper	Paper contribution	Mythology and algorithms used	Dataset used	Results	Drawbacks
	environment was carried out. This study aimed to optimize ransomware detection and classification by deploying a harmony search algorithm. It brought strong abilities and capacities for detecting hidden ransomware.	ransomware detection technique. This method involves learning an enthusiasm-based feature subset selection process. It also deployed Graph CNN to classify ransomware using various numbers of epochs.	ransomware samples and 420 goodware samples.	elements: Accuracy:92.86% Precision:92.86% Recall:92.84% F1-sore:86.03%	
[31], 2023	An AE-based anomaly detection study has been conducted. This study developed a new approach for anomaly detection in cloud networks using the reconstruction error method. It also improved classification performance by deploying multi-classifiers with a hierarchical structure to all classes.	This study used a reconstruction error method based on the AE model. It focused on reconstruction error across all input features, while existing studies applied reconstruction error to a single value.	To evaluate this study, they used a CIDDS-001 dataset, which comprises unidirectional Net Flow data. This dataset includes var attributes such as IP addresses, protocols, bytes, and class labels.	The proposed model detected intrusion with high performance. Accuracy:99.90% Precision:99.58% Recall:99.83% F1-sore:99.91%	Even with high performance in intrusion detection, this study is efficient in the small business environment using OpenStack.
[32], 2023	A study based on ransomware detection improvement has been conducted. This study helped minimize the impacts that crypto-ransomware attacks were causing during the COVID-19 pandemic. The model detected ransomware attacks before they could execute on the network.	The study deployed static analysis which involved converting the PE header into color images and detecting malicious code using Xception CNN and testing method for zero-day ransomware detection.	To evaluate this study, they used two datasets. The first comprises 1000 ransomware samples and 1000 benign while the second contains 1023 ransomware from 25 families and 1134 benign.	The detection of ransomware accurately. The first data showed the performance below: Accuracy:93.73% Precision:92.95% Recall:94.64% F1-sore:93.75% For the second dataset, Accuracy:98.20% Precision:95.50% Recall:98.76% F1-sore:98.12%	The study dealt with crypto-ransomware attacks. However, the researcher used static analysis, which has limitations in detecting heavily obfuscated attacks. To combat these attacks effectively, dynamic analysis or hybrid analysis is necessary.
[33], 2024	This study enhanced ransomware detection and classification using techniques that facilitate feature extraction from the dataset. It also enhances zero-day ransomware detection by identifying instances of ransomware, leading to improved precision rates.	The model deployed stacked AE based on the LSTM model for feature selection using Extreme Gradient Boosting to detect and classify ransomware. The model successfully selected relevant features, which resulted in high accuracy.	The study used a UG Ransomware dataset that comprises 207533 features with 58491 redundant patterns.	Accuracy:95.50% Precision:94.50% Recall:97.80% F1-sore:96.10%	The proposed model has performed ransomware detection. However, the used dataset makes it to be effective in capturing variants and complex ransomware attacks.
[34], 2024	This study implemented an earlier zero-day ransomware detection. The developed model can detect attacks before the data encryption. It also remedied the time-consuming, mostly observed in traditional system-based ransomware detection systems by deploying dynamic analysis.	This study deployed self-coding and ZRS methods for feature extraction to learn potential information from both hidden and visible classes. It also used a CNN detection framework for the early detection of zero-day ransomware.	The used dataset is based on a portable executable (PE) header. It comprises 582 ransomware samples and 942 benign.	Accuracy:96.02% Precision:91.49% Recall:98.47% F1-sore:96.31%	The proposed model used an old dataset, and it is compatible with binary classification tasks and is inefficient for ransomware classification based on their families.

As shown in *Table 2*, several studies have been conducted to detect and classify ransomware attacks in industrial IoT and other domains. Most of these studies have focused on combating ransomware propagation. However, with the increasing number of IoT-connected devices, the cyber threat landscape has also expanded. Cybercriminals and adversarial actors continue to develop new evasion techniques, rendering some existing approaches ineffective due to their limitations. This study analyzes five deep learning models commonly used in ransomware detection, identifying best practices and recommendations for improving ransomware detection in IoT environments.

3.Methodology

3.1Research design

This section explains the methods and approaches used to examine the structure, benefits, and limitations of GAN, AE, LSTM, Bi-LSTM, and CNN for ransomware detection in IIoT, highlighting best practices based on performance evaluation. A combined approach utilizing opcodes and high-order n-grams was implemented to develop a more robust and accurate ransomware detection system capable of capturing various malicious behaviors. Additionally, the principal component analysis (PCA) technique was incorporated to reduce computational complexity by lowering data dimensionality and improving manageability. Finally, descriptive, comparative, and prescriptive analysis methods were employed to analyze and compare our deep learning models.

In our study, opcodes have been emphasized significantly for ransomware pattern detection. Our technique involved extracting opcodes from executable files and analyzing their behaviors using statistical methods. The analysis focused on key aspects such as opcode frequency, unusual opcode patterns, and encryption routine sequences, which are commonly utilized by ransomware. In our study, opcode sequences represented by $S = \{s_1, s_2, s_3, \ldots s_n\}$ were extracted and converted into a human-readable format. The extraction process can be formalized through a mapping function $\mathbb{D}: B \rightarrow S$ where B represents binary input and S opcode sequences results. To capture the characteristics of these aspects, we calculated the frequency distribution of each opcode into a sequence that acts as a foundation for constructing feature vectors that represent ransomware samples. The frequency can be expressed as shown in Equation 1.

$$f(S) = (w_1, w_2, \ldots, w_m) \text{ where } W_k = \frac{\sum_{i=1}^{n} \mathbb{1}(s_i = k)}{n} \quad (1)$$

Where *Wk* represents the frequency of opcode k within a sequence; $\mathbb{1}(s_i = k)$ indicator function returns 1 if s_i corresponds to opcode k and 0 if not, while n is the number of opcodes in a sequence. To reinforce our ransomware detection system, high-order n-grams which are contiguous sequences of multiple items, have been associated. This technique involves capturing more context and detailed patterns in the binary code of executable files, making distinguishing benign and malicious codes easy. The probability of a specific n-gram occurring in a sequence can be expressed as shown in Equation 2.

$$P(g_j|S) = = \frac{\sum_{i=1}^{n-n+1} \mathbb{1}(s_i^n = g_j)}{n - n + 1} \quad (2)$$

Where g_j is a specific n-gram and s_i^n is the subsequence of length n starting at position i within *S.* The feature extraction has been performed by transforming binary code into a suitable format and the feature results are presented as shown in Equation 3:

$$X = \{f(S_1, S_2, S_3, \ldots\ldots S_N) \quad (3)$$

where N represents the number of total ransomware binaries

GAN has been chosen to reinforce ransomware detection and its variants that evade traditional methods. The implementation of the AE model in this study generated file access sequences crucial for efficiently classifying ransomware variants. The LSTM model was selected to address vanishing gradient challenges and effectively handle data sequences of varying lengths. This model retains relevant information while filtering out noise, making it highly efficient for anomaly detection. To prevent malicious code from evading detection, bidirectional data processing and analysis were employed using the Bi-LSTM model. Finally, CNN was utilized to automatically learn spatial hierarchies of features and analyze data representations, enhancing its ability to detect obfuscated or polymorphic malware. *Figure 1* presents the architecture used in this study.

Figure 2 illustrates the working mechanism of ransomware architecture in this study. The proposed architecture consists of three main phases. Phase 1 involves data preprocessing, feature extraction, and selection using opcode and high-order n-gram techniques. After assembling samples in the dataset, data suitability for dynamic analysis processes is ensured by handling missing values, encoding categorical variables, and scaling features. Once meaningful features are extracted from raw data, the

dataset is split into training and testing sets, with 80% allocated for training and 20% for testing. A cross-validation process is applied to mitigate overfitting by iteratively splitting the data into training sets. In Phase 2, techniques are implemented to combine significant patterns that capture the characteristics of ransomware from both approaches. Finally, in Phase 3, the prediction process classifies ransomware based on its signature, distinguishing between attacks and benign traffic.

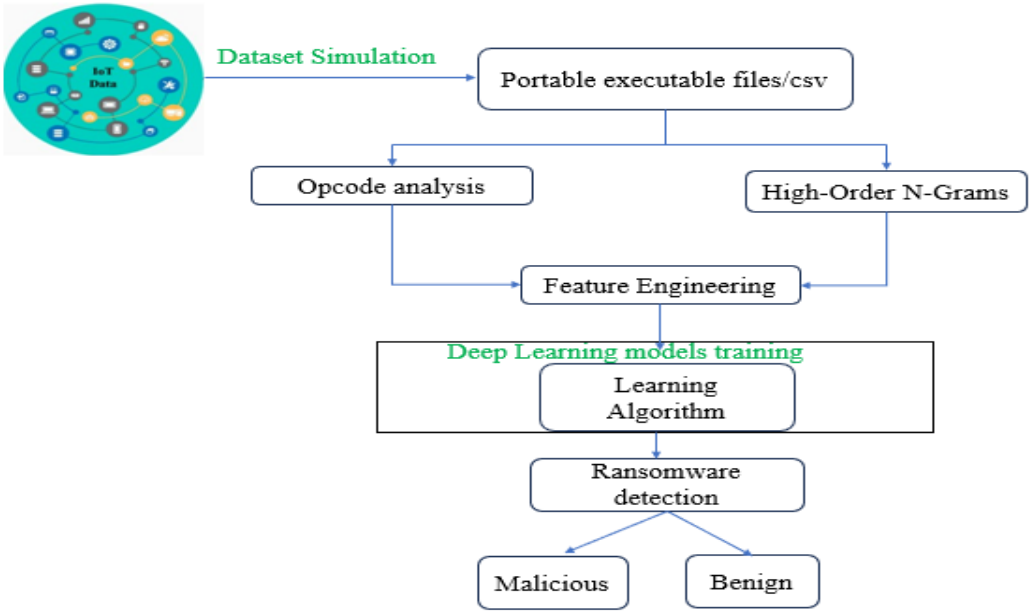

Figure 1 Ransomware detection architecture used in this study

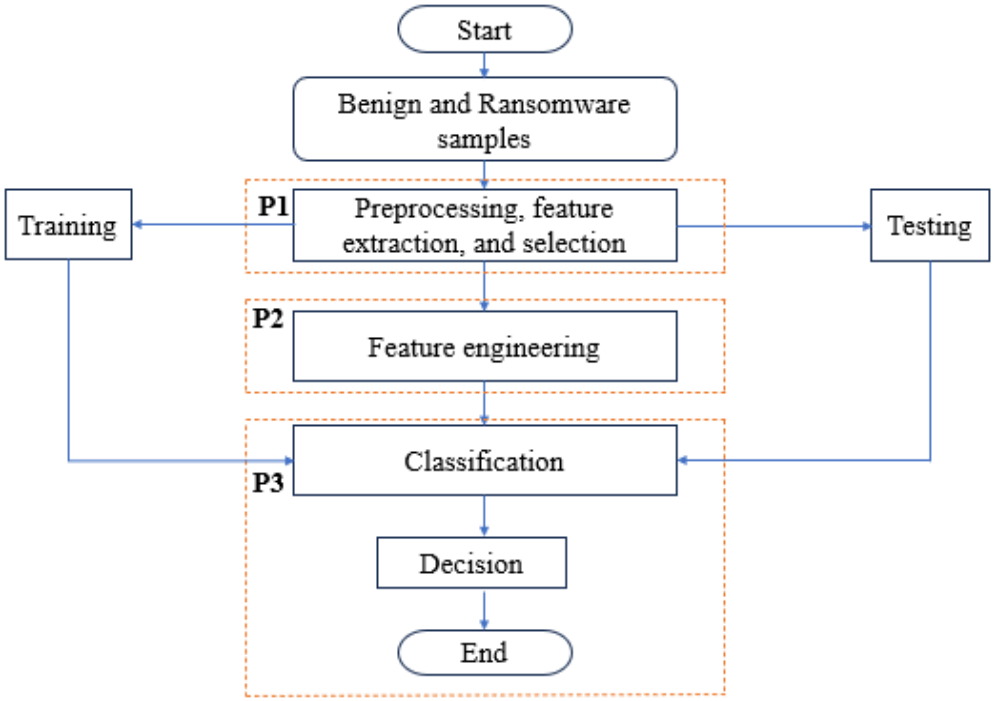

Figure 2 Flowchart of the ransomware detection process using ML

The following outlines the steps used in the algorithm to identify ransomware behavior based on key features and sequences.

Algorithm 1: Feature selection and sequence detection

1: Input: Opcode frequency vector X = {f(S_1), f(S_2), f($S_{3,}$) ……...f(S_N)}, Label: y = ($y_{1,}$ y_2, $y_{3,}$ ……y_N)
2: Output: Predicted label y
3: Initialize models
4: Train models on (X,y)
5: Select top-k features $X_k \subset X$
6 for each sample $S_i \in X_k$ do
7: Reshape S_i into sequence S_i = ($s_{1,}$ $s_{2,}$ ……s_t)
8: Feed S_i into models (proposed models)
9: end for
10: Compute final classification
11: Return: Predicted label

3.2Materials used

To achieve the objectives of this study, both hardware and software tools were utilized. Python version 3.12 (64-bit), a high-level interpreted programming language with dynamic capabilities, was used for coding and simulations. The implementation was carried out on a personal computer with the following specifications: Microsoft Windows 11 operating system, 16 GB of RAM, and an 11th Generation Intel® Core™ i5 processor.

3.2.1Descriptive comparison of our study

Generative adversarial networks

GAN is an unsupervised learning model that identifies and learns patterns or regularities from input data and generates new samples that resemble those in the dataset. It has two main components: a generator generates new samples, and a discriminator distinguishes real data from the generated data [35]. During the training process, the discriminator gets fake samples generated by the generator without being aware of the data source. The discriminator predicts if the input data is real or fake. It calculates the loss and produces signals to the generator to update its weights through the backpropagation phase while discriminator losses stay static. The generator and discriminator have to be trained independently: the generator to minimize the loss while the discriminator is to maximize the loss, as shown in Equation 4:

$$\min\max V(D, G) = Ex\sim pdata(x)[\log(D(x))] + Ez\sim pz(Z)[\log(1-D(G(z)))] \quad (4)$$

GANs function by leveraging neural network principles and utilizing a training set as input to generate new, similar data [36]. They are applicable in various domains, including text-to-image generation, video prediction, object generation, clothing translation, anomaly detection, medicine, language synthesis, and 3D modeling [37]. GANs have several subtypes, such as Conditional GANs, Deep Convolutional GANs, Cycle GANs, and Vanilla GANs, among others [38]. *Table 3* summarizes the advantages and limitations of GANs.

Table 3 Advantages of limitations of GANs

Advantages	Limitations
-Unsupervised models can train themselves after getting initial input and can learn from unlabeled data -Capable of identifying anomalies based on data modeling measurements of the generator and discriminator. -They have realistic sample creation capacities	-Not effective in training large varied and advanced datasets. -They struggle to evaluate results from complex given tasks. -Suffering from mode collapse or learning to produce only output because of their high plausibility and ability to trick discriminators. -As the generator improves during training, GANs may not converge because the discriminator struggles to differentiate real and fake data.

Autoencoders (AE)

AE are unsupervised deep learning models that encode input data into a compressed, meaningful representation and then decode it to reconstruct the original data with minimal loss [39]. These neural networks are highly effective due to their ability to capture complex nonlinear correlations. Autoencoder models have played a significant role in various domains, including cybersecurity. They can learn a fundamental representation of normal data and reconstruct it with minimal error, making them useful for intrusion detection by analyzing network traffic patterns.

In the IoT domain, AE are particularly valuable for detecting anomalies within large-scale, sparse network sensors. They can identify irregularities in sensing data, device malfunctions, environmental anomalies, and security breaches in smart systems.

Autoencoder architecture comprises three components: encoder, decoder, and bottleneck. The encoder has input layers responsible for taking raw

data and hidden layers that deal with the dimensionality reduction of input, capturing important features and patterns. Bottleneck layers that handle significant dimensionality reduction limit the encoder's layers. On the decoder side, the operations performed by the layers are the inverse of those performed by the encoder to produce a reconstructed output that should be identical to the input data.

Advantages and limitations of AE

Autoencoder-based deep learning models are highly effective in learning feature representations by uncovering intrinsic data structures, capturing discriminative information, and generating more robust image representations. They provide effective regularization techniques and serve as powerful tools for detecting malicious activities in networked devices, as well as identifying faults and security breaches in smart systems.

However, these models may face challenges such as limited data availability and complex architectures, which can lead to overfitting in hidden unit layers. Additionally, tuning parameters can significantly impact the quality of learned representations.

LSTM

LSTM is a recurrent neural network (RNN) type designed to process and analyze sequential data while capturing dependencies in prediction tasks. It is an advanced deep-learning model that addresses the vanishing gradient problem, overcoming a key limitation of traditional RNNs [40].

LSTM is a powerful tool due to its ability to retain and utilize order sequence dependencies, making it highly effective for solving various problems. It is widely applied in domains such as machine translation, speech recognition, malware detection and classification, language modeling, and video analysis [41].

The LSTM architecture comprises memory cells, input, output, and forget gates. In the LSTM architecture, memory cells store data over extended periods, while gates control the flow of information in and out of the cells. Input gates determine internal state updates based on the previous internal state [42]. The forget gates help to determine the number of previous internal states that should be forgotten, while output gates regulate the influence of the internal states. Nevertheless, the LSTM gates are calculated using Equations 5 to 7:

$$I_t=\sigma(X_tW_{xi}+H_{t-1}W_{hi}+b_i) \quad (5)$$

$$F_t=\sigma(X_tW_{xf}+H_{t-1}W_{hf}+b_f) \quad (6)$$

$$O_t=\sigma(X_tW_{xO}+H_{t-1}W_{hO}+b_O) \quad (7)$$

Where W_{xi}, W_{xf}, W_{xO} $\epsilon\mathbb{R}^{d\ x\ h}$ and W_{hi}, W_{hf}, W_{hO} $\epsilon\mathbb{R}^{h\ x\ h}$ are weight parameters, b_i, b_f, b_O $\epsilon\mathbb{R}^{1\ x\ h}$ are bias parameters and H_{t-1} $\epsilon\mathbb{R}^{n\ x\ h}$ are the hidden state of the previous time step.

Advantages and limitations of LSTM

LSTM networks offer several advantages that enhance their effectiveness across various applications. They excel at learning long-term dependencies in sequential data, a challenge that traditional neural networks struggle with due to vanishing gradient issues. Additionally, LSTMs can selectively retain relevant information while filtering out noise, ensuring more accurate processing. Their ability to handle data sequences of varying lengths further enhances their versatility in diverse applications. In noisy and missing values management, it is efficient for large datasets because of its capabilities of being trained effectively through backpropagation through time (BPTT). However, LSTM has some limitations: It processes input data only in a forward direction, which can be a barrier to good effectiveness. The forget gate can also disturb the classification performance, and it may not be significant as it is usually open to allow information to pass through.

Bidirectional LSTM

Bi-LSTM is an extended version of the LSTM model designed to enhance performance. The key difference between Bi-LSTM and LSTM is that while LSTM processes input data in a single direction, Bi-LSTM processes it in both forward and backward directions, addressing one of LSTM's limitations [43, 44].

In Bi-LSTM, the forward layer processes input data from left to right (indicated by the green arrow), while the backward layer processes it in the opposite direction. Bi-LSTM is an effective tool for ransomware and anomaly detection due to its ability to process data in both directions and capture malicious activity. It can learn from the patterns of application programming interface calls (API) sequences and analyze others' behaviors of the systems [45]. It is good for abdominal classification, like ransomware attacks, and has achieved an accuracy of 98.91% [46].

Convolutional neural network model

CNN is a type of deep learning used in various tasks like image recognition and classification, a powerful tool of cybersecurity used to learn features and detect malware and other anomalies. They are also robust for data variability, which makes them suitable for analyzing complex and evolving structures of ransomware. CNN architectures comprise five layers: convolutional layer, pooling layer, fully connected layer, and fully connected input and output layer, where each layer has its specific functions [47].

Convolutional layer: A convolutional layer is the main component of CNN models that applies a convolution operation to input data. It extracts specific features from the input and transforms them into a smaller output volume. The convolutional layer consists of learnable kernels, represented as square matrices (e.g., 3×3 or 5×5), that facilitate input feature mapping. These kernels slide over regions of the feature map, performing convolution operations to extract spatial patterns and key features [48].

Pooling layer: A pooling layer is an important component of CNN that follows the convolutional layer. It reduces special dimensions, ultimately decreasing the network's complexity. The pooling layer enables recognition of objects even when their shapes are distorted or viewed from different angles [49].

Fully connected layers: The last layer of a CNN is known as the fully connected layer because each neuron in one layer is connected to every neuron in the preceding layer and acts as a classifier that enables the predictions [50].

Advantages and limitations

CNNs are highly effective at extracting features, automatically learning spatial hierarchies of features, and analyzing data representations, which enhance their ability to detect obfuscated or polymorphic malware. However, CNN's detection of anomalies requires a large amount of labeled training data, which makes it time-consuming. It also struggles to capture temporal dependencies and may overfit when using a small trained dataset.

3.3Experimentation

Dataset description

This study utilizes ransomware PE header features and the IoT-23 dataset for analysis. The ransomware PE header dataset comprises 2,157 samples categorized into 25 ransomware families, including 1,134 benign samples and 1,023 well-known ransomware samples. It contains raw data extracted from the PE header, specifically the first 1,024 bytes of each binary.

The IoT-23 dataset consists of 20 malware captures extracted from various IoT network traffic sources, along with three benign samples. Developed by the Avast AIC laboratory, this dataset provides a curated collection of labeled data, enhancing security research through ML-based threat detection.

Used parameters for deep learning models

Table 4 presents the various parameters used for implementing the models in this study.

Table 4 Configuration parameters for used models

Parameters	GAN	AE	LSTM	Bi-LSTM	CNN
Optimizer	Adam	Adam	Adam	Adam	Adam
Layers	3	2	3	2	2MaxPool, 1Flatten, 3Conv2D
Activation function	Rectified linear unit (ReLu) & Sigmoid	ReLu & Sigmoid	ReLu & Sigmoid	Sigmoid	ReLu &Sigmoid
Number of Epoch	1500	1200	100	100	100
Batch size	64	64	32	32	32
Loss function	Binary cross entropy	Binary cross entropy	Binary cross entropy	Binary cross entropy	Binary cross entropy
Learning rate	0.0002	0.001	0.02	0.01	0.4

4.Results

4.1 Effective analysis-based hyperparameters

This section presents the analysis results of the used models, focusing on their hyperparameters and operational requirements. Hyperparameters, such as the optimizer, learning rate, activation function, and number of hidden layers, define the learning process of each model (*Table 5*).

Table 5 Hyperparameters used for different models

Hyperparameters	GANs	AE	LSTM	CNN
Best Optimizer	Adam	Adam	Stochastic gradient Descent	Adam, SGD with custom parameters
Best Learning rate	3e-4	3e-4	1e-4 to 1	3e-4, 1e-4 to1
Regularization methods	Weight decay	Sparsity, weight decay	Weight decay	Weight decay, dropout, early stopping, data segmentation
Number of hidden layers	3	3	2-3 mostly used	3
Activation function	ReLU (G), Sigmoid (D)	ReLU, Sigmoid	Sigmoid, Tanh	ReLU
Number of Epochs	≥200	≥900	≥1000	≥2048
Batch size	≥64	≥512	64	32 r 64

4.2 Results of the deep learning model's metrics-based confusion matrix data

A confusion matrix is a performance evaluation tool that assesses a model's effectiveness. It categorizes predictions into correctly classified instances, which can be either true positives or true negatives, and misclassified instances, such as false positives and false negatives. The following Equations 8 to 11 illustrate these classifications.

$$\text{Accuracy} = \frac{TN+TP}{TP+TN+FN+FP} \quad (8)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (9)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (10)$$

$$\text{F1-score} = 2\times \frac{Precision*Recall}{Precision+Recall} \quad (11)$$

Table 6 presents the experimental results evaluating the performance of deep learning models using the dataset. The experiments were conducted using ReLU and Sigmoid activation functions, compiled with the Adam optimizer, and trained for 100 epochs with a batch size of 32. The results indicate that GANs exhibited lower performance, whereas CNN achieved superior accuracy, precision, and F1-score compared to the other models. *Table 7* presents the evaluation of five deep-learning models using the IoT-23 dataset. The results consistently demonstrate the superior performance of the CNN model compared to the others.

Figures 3, *4*, *5*, *6* and *7* present the performance evaluation results of the deep learning models used in this study.

Table 6 Experimental results of deep learning models using the ransomware PE header feature dataset

Used model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
GAN	89.52	84.86	93.84	93.84
AE	87.36	89.47	86.95	89.47
LSTM	96.29	95.40	95.40	95.84
CNN	96.98	97.10	96.87	96.98
Bi-LSTM	95.52	94.67	97.00	95.82

Table 7 Evaluation of deep learning models using the IoT-23 dataset

Used model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
GAN	89.16	92.90	93.61	93.25
AE	88.92	93.30	92.23	92.76
LSTM	96.35	97.44	94.82	96.11
CNN	96.25	97.50	95.62	96.55
Bi-LSTM	95.00	94.91	93.87	94.38

The receiver operating characteristic - area under the curve (ROC-AUC) curve in *Figure 8* compares the performance of five deep learning models—CNN, LSTM, Bi-LSTM, GAN, and AE—based on their true positive rate (TPR) against the FPR. The CNN model (green) shows the best performance, achieving the highest TPR across all thresholds, followed by LSTM and Bi-LSTM. GAN and AE exhibit lower performance, as indicated by their lower TPR values.

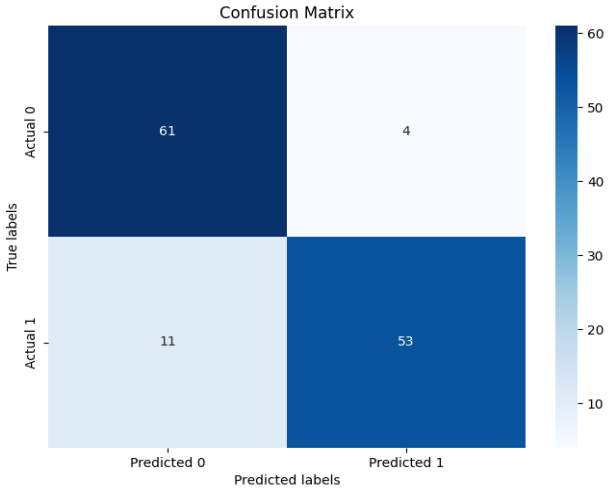

Figure 3 Confusion matrix of GAN model

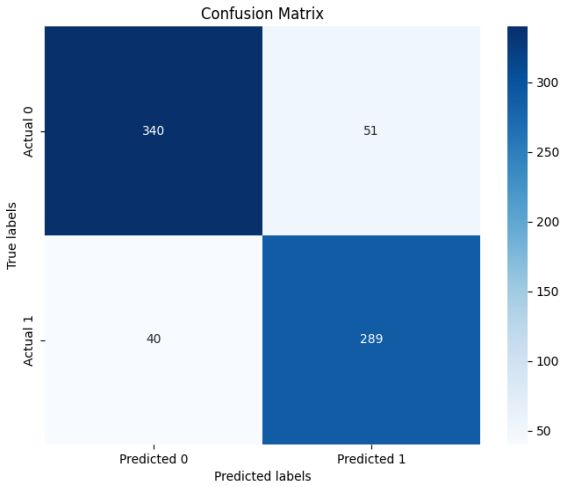

Figure 4 Confusion matrix for AE model

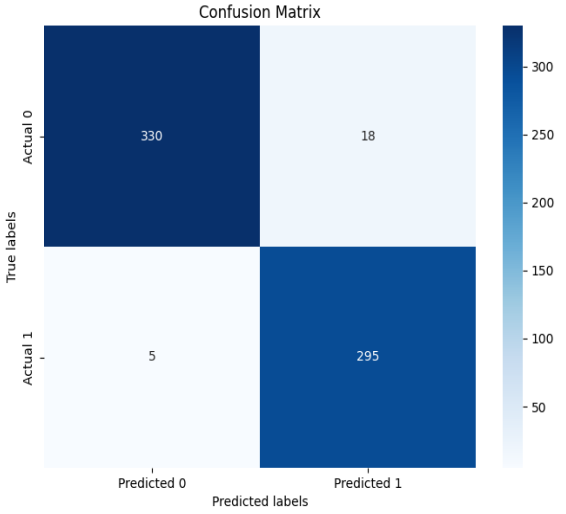

Figure 5 Confusion matrix results for CNN

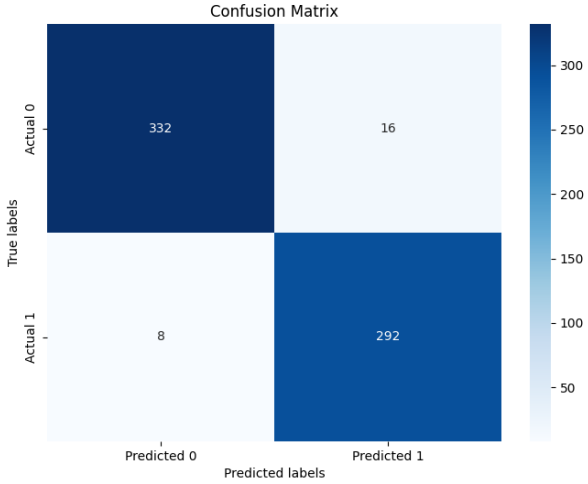

Figure 6 Confusion matrix results for LSTM

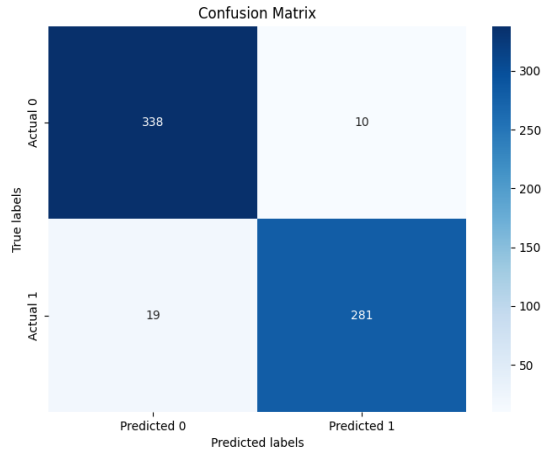

Figure 7 Confusion matrix results for Bi-LSTM

The ROC-AUC curve in *Figure 8* compares the performance of five deep learning models—CNN, LSTM, Bi-LSTM, GAN, and AE—based on their TPR against the FPR. The CNN model (green) shows the best performance, achieving the highest TPR across all thresholds, followed by LSTM and Bi-LSTM. GAN and AE exhibit lower performance, as indicated by their lower TPR values.

5.Discussion

This study utilized two datasets and observed different results after evaluating deep learning models. In the first phase of model training with the ransomware PE header feature dataset, the results demonstrated higher performance compared to those obtained with the IoT-23 dataset. The study aimed to identify which deep learning models are most

effective in detecting ransomware attacks targeting IIoT.

In both cases, GAN and AE models did not achieve high accuracy compared to other models. In the first experiment using ransomware PE header features, the confusion matrix results proved that the model accurately detected correct predictions at 89.16%, which means 114 among 128 samples (*Figure 3*) and incorrectly predicted 4 samples as negative outcomes and 11 as positive outcomes. AE, as shown in *Figure 4*, correctly detected 629 among 720 samples as good outcomes, equivalent to 87.36%, and produced negative outcomes in 91 samples. However, as shown in *Figure 5*, CNN demonstrated its superior performance by detecting correctly 625 among 648 samples, equivalent to 96.98%, and gave 26 samples of false negatives and positives. *Figures 6* and *7* illustrate the performance of LSTM and Bi-LSTM, where LSTM correctly detected 624 samples (96.29%) and gave incorrect outcomes on 24 samples, while Bi-LSTM accurately performed at 95.52% with a failure of 4.48%. Similarly, in the second experiment using the IoT-23 dataset as shown in *Table 7*, GAN and AE models again exhibited lower accuracy of 89.16% and 88.92%, whereas LSTM and CNN achieved 96.35% and 96.89%, respectively. Based on these confusion matrix results, we can confirm that GAN and AE have been challenged to distinguish ransomware and benign based on their opcode sequences, which present security risk-based misclassification of benign activities and missed detection that lead to unnecessary panic and operational disruption, allowing threats to go undetected and potentially cause several damages. The confusion matrix results confirmed the security effectiveness of CNN, LSTM, and Bi-LSTM by detecting ransomware instances, which is crucial for harm prevention. However, improvements in these models are required to fight against the missed detection rate.

The CNN, LSTM, and Bi-LSTM models provided performance different from others based on several factors. CNNs are easier to train and capable of extracting spatial features from data, making them more reliable for detecting new and unseen ransomware variants while also being computationally efficient. LSTMs are highly effective in sequential data analysis, excelling in examining opcodes within executable files, system logs, and network traffic, and learning temporal dependencies and patterns over time. In contrast, GANs and AEs do not inherently capture temporal dependencies in sequential data, which affects their performance in the datasets used. AE models encode and decode data independently, making it difficult to capture long-range dependencies, as reflected in their lower accuracy in the first dataset. Additionally, GANs require simultaneous training of their components, which slows down the detection process for malicious code. Although GANs can generate diverse samples, they may not generalize to new data for classification tasks, unlike CNNs and LSTMs. However, GANs are beneficial when working with less labeled data, while AE models require a large amount of unlabeled data, contributing to variations in model performance across different datasets. *Table 8* presents a comparison of ransomware detection performance between this study and previous research.

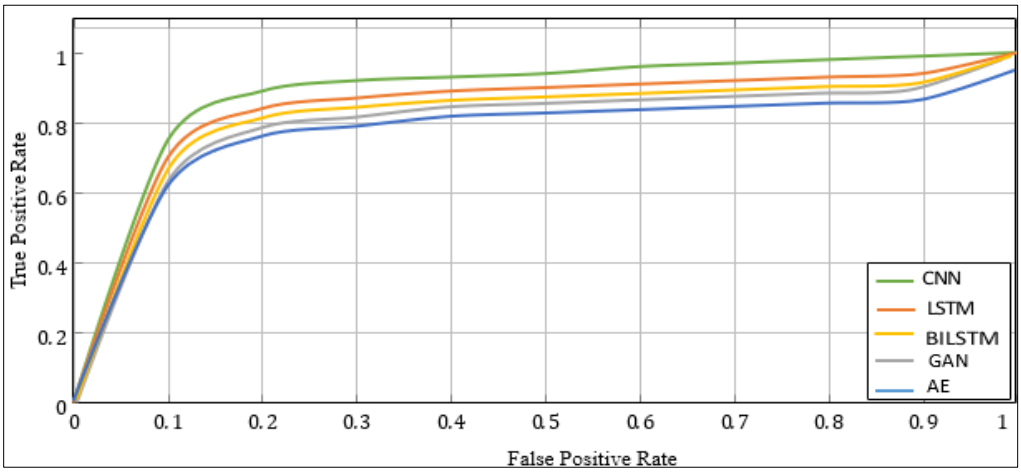

Figure 8 ROC-AUC curve comparison for deep learning models

Table 8 Comparison with previous ransomware studies

Study	Feature type	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
[51], 2021	PE	91.13	90.76	92.06	91.40
[29], 2022	Image	84.01	85.06	80.00	81.02
[32], 2023	PE Header	93.73	92.95	94.64	93.75
[34], 2024	PE Header	92.02	91.49	98.47	96.31
[33], 2024	PE Header	95.50	94.50	97.80	96.10
Our study	PE Header	96.89	96.66	95.98	96.55

Limitations

Our study requires continuous updates to enable models to detect new and unseen ransomware variants that exhibit different behaviors. It requires an increased processing time and resource constraints, especially when dealing with large datasets. Its high false positives and negatives could impact the reliability of the system. Several factors must be considered when deploying these deep learning models in IoT environments. Researchers should consider computational requirements that may affect model performance. For instance, GAN model elements are required to be trained simultaneously when deployed in an IoT environment. However, this process can require significant computational power and memory when dealing with large datasets. AE models require substantial resources, especially for complex architectures, while LSTM models are computationally intensive because of their complex architecture and the need for state information maintained over long sequences. CNNs also require significant computational resources when dealing with high-resolution inputs, training, and inference. Researchers are suggested to apply techniques that could minimize these computational requirements, such as compression, hardware accelerators, and efficient architecture for better-suiting resource constraints of IoT devices. A complete list of abbreviations is listed in *Appendix I.*

6.Conclusion

In this study, GANs, AE, LSTM, Bi-LSTM, and CNN were examined based on their structure, benefits, and limitations to identify the most suitable model for real-time ransomware detection in an industrial IoT environment. Opcode sequences associated with high-order n-grams were deployed for accurate malicious code detection, along with descriptive, comparative, and prescriptive analysis methods to analyze, evaluate, and compare model performance using the ransomware PE header feature and IoT-23 datasets. The evaluation results demonstrate the superior performance of LSTM and CNN models, where LSTM achieved an accuracy of 95.89% using the PE header feature dataset and 96.35% with the IoT-23 dataset, while CNN achieved 96.98% accuracy with the PE header feature dataset and 96.89% using the IoT-23 dataset. However, given the continuously evolving attack tactics of cybercriminals, relying on a single-model-based malware detection approach is insufficient. A robust hybrid model utilizing an updated dataset should be deployed to detect attacks before execution. Additionally, as GAN and AE models can detect malicious activities and perform well even in the presence of normalcy deviations, improved regularization techniques are required to prevent overfitting in these models.

Acknowledgment

None.

Conflicts of interest

The authors have no conflicts of interest to declare.

Data availability

The datasets used in this study are publicly available and can be accessed at the following links:
Ransomware PE header feature dataset: https://data.mendeley.com/datasets/p3v94dft2y/2.
IoT-23 dataset: https://www.stratosphereips.org/datasets-iot23.

Author's contribution statement

Deo Irankunda: Background work, conceptualization, methodology, investigation, data collection, analysis, implementation, and writing original draft manuscript. **Khalid El Fazazy & Tairi Hamid:** Analysis and review of draft manuscript. **Jamal Riffi:** Supervision, challenges investigation, and review work.

References

[1] Malik PK, Sharma R, Singh R, Gehlot A, Satapathy SC, Alnumay WS, et al. Industrial internet of things and its applications in industry 4.0: state of the art. Computer Communications. 2021; 166:125-39.

[2] Peter O, Pradhan A, Mbohwa C. Industrial internet of things (IIoT): opportunities, challenges, and requirements in manufacturing businesses in emerging economies. Procedia Computer Science. 2023; 217:856-65.

[3] Jhanjhi NZ, Humayun M, Almuayqil SN. Cyber security and privacy issues in industrial internet of

things. Computer Systems Science & Engineering. 2021; 37(3):361-80.
[4] Humayun M, Jhanjhi NZ, Alsayat A, Ponnusamy V. Internet of things and ransomware: evolution, mitigation and prevention. Egyptian Informatics Journal. 2021; 22(1):105-17.
[5] Alraizza A, Algarni A. Ransomware detection using machine learning: a survey. Big Data and Cognitive Computing. 2023; 7(3):1-24.
[6] Jose J, Jose DV, Rao KS, Janz J. Impact of machine learning algorithms in intrusion detection systems for internet of things. In international conference on advances in computing and communications 2021 (pp. 1-6). IEEE.
[7] https://cybersecurityventures.com/global-ransomware-damage-costs-predicted-to-reach-250-billion-usd-by-2031/. Accessed 25 January 2025.
[8] Kalnoor G, Gowrishankar S. Markov decision process based model for performance analysis an intrusion detection system in IOT networks. Journal of Telecommunications and Information Technology. 2021; (3):42-9.
[9] Nabi AU, Ahmed M, Abro A. An overview of firewall types, technologies, and functionalities. International Journal of Computing and Related Technologies. 2022; 3(1):10-6.
[10] Al-hawawreh M, Alazab M, Ferrag MA, Hossain MS. Securing the industrial internet of things against ransomware attacks: a comprehensive analysis of the emerging threat landscape and detection mechanisms. Journal of Network and Computer Applications. 2024; 223:103809.
[11] Khalil RA, Saeed N, Masood M, Fard YM, Alouini MS, Al-naffouri TY. Deep learning in the industrial internet of things: potentials, challenges, and emerging applications. IEEE Internet of Things Journal. 2021; 8(14):11016-40.
[12] Demertzi V, Demertzis S, Demertzis K. An overview of privacy dimensions on the industrial Internet of Things (IIoT). Algorithms. 2023; 16(8):1-32.
[13] Muñoz DC, Valiente AD. A novel botnet attack detection for IoT networks based on communication graphs. Cybersecurity. 2023; 6(1):1-17.
[14] Serror M, Hack S, Henze M, Schuba M, Wehrle K. Challenges and opportunities in securing the industrial internet of things. IEEE Transactions on Industrial Informatics. 2020; 17(5):2985-96.
[15] https://www.cisa.gov/sites/default/files/FactSheets/NCCIC%20ICS_FactSheet_WannaCry_Ransomware_S508C.pdf. Accessed 25 January 2025.
[16] Gerodimos A, Maglaras L, Ferrag MA, Ayres N, Kantzavelou I. IoT: communication protocols and security threats. Internet of Things and Cyber-Physical Systems. 2023; 3:1-13.
[17] Jaloudi S. Communication protocols of an industrial internet of things environment: a comparative study. Future Internet. 2019; 11(3):1-18.
[18] Younan M, Houssein EH, Elhoseny M, Ali AA. Challenges and recommended technologies for the industrial internet of things: a comprehensive review. Measurement. 2020; 151:107198.
[19] Chalapathi GS, Chamola V, Vaish A, Buyya R. Industrial internet of things (IIoT) applications of edge and fog computing: a review and future directions. Fog/edge Computing for Security, Privacy, and Applications. 2021: 293-325.
[20] Vehabovic A, Ghani N, Bou-harb E, Crichigno J, Yayimli A. Ransomware detection and classification strategies. In international black sea conference on communications and networking (BlackSeaCom) 2022 (pp. 316-24). IEEE.
[21] Benaddi H, Jouhari M, Ibrahimi K, Ben OJ, Amhoud EM. Anomaly detection in industrial IoT using distributional reinforcement learning and generative adversarial networks. Sensors. 2022; 22(21):1-18.
[22] Andrade ED, Viterbo J, Vasconcelos CN, Guérin J, Bernardini FC. A model based on LSTM neural networks to identify five different types of malware. Procedia Computer Science. 2019; 159:182-91.
[23] Zahoora U, Khan A, Rajarajan M, Khan SH, Asam M, Jamal T. Ransomware detection using deep learning based unsupervised feature extraction and a cost sensitive pareto ensemble classifier. Scientific Reports. 2022; 12(1):1-15.
[24] Khan F, Ncube C, Ramasamy LK, Kadry S, Nam Y. A digital DNA sequencing engine for ransomware detection using machine learning. IEEE Access. 2020; 8:119710-9.
[25] Riaz S, Latif S, Usman SM, Ullah SS, Algarni AD, Yasin A, et al. Malware detection in internet of things (IoT) devices using deep learning. Sensors. 2022; 22(23):1-22.
[26] Khan SH, Alahmadi TJ, Ullah W, Iqbal J, Rahim A, Alkahtani HK, et al. A new deep boosted CNN and ensemble learning based IoT malware detection. Computers & Security. 2023; 133:1-14.
[27] Pavithra J, Selvakumara SS. A comparative study on detection of malware and benign on the internet using machine learning classifiers. Mathematical Problems in Engineering. 2022; 2022(1):1-8.
[28] Saran N, Kesswani N. A comparative study of supervised machine learning classifiers for intrusion detection in internet of things. Procedia Computer Science. 2023; 218:2049-57.
[29] Barros PH, Chagas ET, Oliveira LB, Queiroz F, Ramos HS. Malware-SMELL: a zero-shot learning strategy for detecting zero-day vulnerabilities. Computers & Security. 2022; 120:102785.
[30] Khalid AH, Mahmood K, Khalid M, Othman M, Al DM, Osman AE, et al. Optimal graph convolutional neural network-based ransomware detection for cybersecurity in IoT environment. Applied Sciences. 2023; 13(8):1-17.
[31] Torabi H, Mirtaheri SL, Greco S. Practical autoencoder based anomaly detection by using vector reconstruction error. Cybersecurity. 2023; 6(1):1-13.
[32] Moreira CC, Moreira DC, De SJCD. Improving ransomware detection based on portable executable

header using xception convolutional neural network. Computers & Security. 2023; 130:103265.
[33] Nkongolo MN, Tokmak M. Ransomware detection using stacked autoencoder for feature selection. Indonesian Journal of Electrical Engineering and Informatics. 2024; 12(1):142-70.
[34] Cen M, Deng X, Jiang F, Doss R. Zero-ran sniff: a zero-day ransomware early detection method based on zero-shot learning. Computers & Security. 2024; 142:1-14.
[35] Bennmarker G. Exploring GANs to generate attack-variations in IoT networks. Thesis, Uppsala University. 2023.
[36] Kc B, Sapkota S, Adhikari A. Generative adversarial networks in anomaly detection and malware detection: a comprehensive survey. Advances in Artificial Intelligence Research. 2024; 4(1):18-35.
[37] Alqahtani H, Kavakli-thorne M, Kumar G. Applications of generative adversarial networks (GANs): an updated review. Archives of Computational Methods in Engineering. 2021; 28:525-52.
[38] Goodfellow I, Pouget-abadie J, Mirza M, Xu B, Warde-farley D, Ozair S, et al. Generative adversarial networks. Communications of the ACM. 2020; 63(11):139-44.
[39] Pinaya WH, Vieira S, Garcia-dias R, Mechelli A. Autoencoders. In machine learning 2020 (pp. 193-208). Academic Press.
[40] Maniath S, Ashok A, Poornachandran P, Sujadevi VG, AU PS, Jan S. Deep learning LSTM based ransomware detection. In recent developments in control, automation & power engineering 2017 (pp. 442-6). IEEE.
[41] Kumar A, Bhatia A, Kashyap A, Kumar M. LSTM network: a deep learning approach and applications. In advanced applications of NLP and deep learning in social media data 2023 (pp. 130-50). IGI Global.
[42] Altunay HC, Albayrak Z. A hybrid CNN+ LSTM-based intrusion detection system for industrial IoT networks. Engineering Science and Technology, an International Journal. 2023 1; 38:1-13.
[43] Aslan Ö, Yilmaz AA. A new malware classification framework based on deep learning algorithms. IEEE Access. 2021; 9:87936-51.
[44] Avci C, Tekinerdogan B, Catal C. Analyzing the performance of long short-term memory architectures for malware detection models. Concurrency and Computation: Practice and Experience. 2023; 35(6):1-15.
[45] Roy KC, Chen Q. Deepran: attention-based bilstm and CRF for ransomware early detection and classification. Information Systems Frontiers. 2021; 23:299-315.
[46] Alassafi MO, Hasan SH, Badri S, Hasan SH. Optimized Bi-LSTM: a novel approach for attack detection in industrial IoT. Signal, Image and Video Processing. 2024; 18(5):4903-13.
[47] Vakalopoulou M, Christodoulidis S, Burgos N, Colliot O, Lepetit V. Deep learning: basics and convolutional neural networks (CNNs). Machine Learning for Brain Disorders. 2023: 77-115.
[48] Liu C, Cheng F. A survey of image classification algorithms based on graph neural networks. In 3D imaging technologies-multi-dimensional signal processing and deep learning: mathematical approaches and applications, 2021 (pp. 203-12). Springer Singapore.
[49] Ajit A, Acharya K, Samanta A. A review of convolutional neural networks. In international conference on emerging trends in information technology and engineering (ic-ETITE) 2020 (pp. 1-5). IEEE.
[50] Alzubaidi L, Zhang J, Humaidi AJ, Al-dujaili A, Duan Y, Al-shamma O, et al. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. Journal of Big Data. 2021; 8:1-74.
[51] Rezaei T, Manavi F, Hamzeh A. A PE header-based method for malware detection using clustering and deep embedding techniques. Journal of Information Security and Applications. 2021; 60:102876.

Deo Irankunda is a Ph.D. student in Computer Science, specializing in Cybersecurity, at the University of Sidi Mohamed Ben Abdellah (USMBA) in Fes, Morocco. He earned his Master's degree in Embedded and Mobile Systems, specializing in Embedded Systems, from the Nelson Mandela African Institution of Science and Technology (NM-AIST) in Arusha, Tanzania, in 2021. Additionally, he has a background in Telecommunications Engineering and Network and System Security from Hope Africa University in Bujumbura, Burundi. His research interests include the Internet of Things (IoT), Industrial Automation, Information Security, Machine Learning, and Internet Governance.
Email: deo.irankunda@usmba.ac.ma

Dr. Khalid El Fazazy is a Full Professor in Computer Science, specializing in Bioinformatics and Artificial Intelligence. He earned his master’s degree in Bioinformatics and Artificial Intelligence from the National School of Applied Sciences in Tangier, Morocco, and completed his PhD in computer science at the Faculty of Sciences Dhar El Mahraz in Fes, Morocco. His research encompasses several advanced domains, including text mining, medical imaging, bioinformatics, and web semantics. Dr. El Fazazy is particularly focused on advancing deep learning techniques to enhance medical imaging, as well as on projects that emphasize the development of large language models. His work reflects a dedication to pushing the boundaries of technology in both theoretical and applied aspects of computer science.
Email: elfazazy.khalid@usmba.ac.ma

Tairi Hamid received his Ph.D. degree in 2001 from the University of Sidi Mohamed Ben Abdellah, Morocco. In 2002 he did a postdoc in the Image Processing Group of the Laboratory LE2I (Laboratoire d'Electronique, Informatique et Image). Since 2003, he has been an associate professor at the University Sidi Mohamed Ben Abdellah, where he obtained his HDR in 2009. He is currently a PES professor. His main research interests concern visual tracking for robotic control, 3-D reconstruction of Artificial Vision, Medical Images, Visual Information Retrieval, Pattern Recognition, and Machine Learning.
Email: hamid.tairi@usmba.ac.ma

Jamal Riffi is a Professor of Computer Science at Sidi Mohamed Ben Abdellah University in Fez, Morocco. He is a member of the LISAC Laboratory. His areas of expertise include Data Mining and Deep Learning. His primary research interests encompass Text Mining and Cybersecurity, Image Mining and Medical Image Analysis, as well as Bioinformatics.
Email: jamal.riffi@usmba.ac.ma

Appendix I

S. No.	Abbreviation	Description
1	AE	Autoencoders
2	API	Application Programming Interface
3	Bi-LSTM	Bidirectional Long Short Terms Memory
4	BPTT	Backpropagation Through Time
5	CNN	Convolutional Neuron Networks
6	DDoS	Distributed Denial of Services
7	DNA	Deoxyribonucleic Acid
8	DT	Decision Tree
9	FPR	False Positive Rate
10	GAN	Generative Adversarial Network
11	IBM	Internation Business Machines
12	IDS	Intrusion Detection System
13	IIoT	Industrial Internet of Things
14	IoT	Internet of Things
15	IT	Information Technology
16	LSTM	Long Short-Term Memory
17	LRC	Computer Science Lab
18	ML	Machine Learning
19	MQTT	Message Queuing Telemetry Transport
20	NB	Naïve Bayes
21	OT	Operational Technology
22	PCA	Principal Component Analysis
23	PE	Portable Executable
24	ReLu	Rectified Linear Unit
25	RF	Random Forest
26	RNN	Recurrent Neural Network
27	ROC-AUC	Receiver Operating Characteristic - Area Under the Curve
28	SVM	Support Vector Machine
29	TPR	True Positive Rate