Journal of Harbin Engineering University
ISSN: 1006-7043

Vol 45 No. 3
March 2024

A Novel Fuzzing for RPL Network Vulnerability Analysis and Vision Transformer-based Attack Detection for IIoT

Waleed Almuseelem
Faculty of Computing and Information Technology (FCIT), University of Tabuk, Tabuk 47713, Saudi Arabia;

Abstract
Recent scientific advancements in information and communication technologies enable Industry 4.0 (I4.0), which empowers smart manufacturing with unprecedented operational efficiency and productivity. Integrating the smart Industrial Internet of Things (IIoT) facilitates continuous real-time monitoring of manufacturing processes by establishing safety controls through data collection. Despite the substantial benefits, the vast network of interconnected IoT devices in the I4.0 environment is vulnerable to cyber security threats. Routing Protocol for Low Power Lossy Networks (RPL) is a reliable, energy-efficient, and flexible way to set up a routing framework for IIoT-based critical industrial communication infrastructure. However, network security is a critical concern in RPL-based IIoT environments due to complex patterns and subtle deviations in the behavior of the network. Therefore, it is crucial to introduce novel security solutions with more accurate vulnerability analysis and attack detection. This work proposes a Novel RPL Security (NRS) approach that includes fuzzing-based vulnerability analysis and vision transformer-based attack detection to solve the abovementioned issues. The proposed work encompasses two primary components: the Wasserstein Generative Adversarial Network (WGAN)-based fuzzing method for RPL network vulnerability analysis and vision transformer-based attack discovery. In the first method, routing data from the RPL-IIoT network is collected, and the fuzzing model is combined with the WGAN to improve the vulnerability distribution in the fuzzer output data. The analyzed fuzzer output data is converted into images and fed into the vision transformer model for attack discovery. The vision transformer improves attack detection accuracy by effectively capturing complex patterns and subtle deviations in network behavior. Moreover, the efficacy of the proposed NRS is evaluated using Contiki/Cooja-based simulations and Python-based machine-learning models. The results are validated for vulnerability analysis and attack detection using various metrics such as detection accuracy, fuzzer output recognition rate, triggered efficiency of vulnerabilities, and diversity of generated data, revealing the notable outcome of the proposed approach.
Keywords: Industrial 4.0, Industrial Internet of Things, Network Vulnerability Analysis, Novel Fuzzing Method, RPL Routing Security, Deep Learning, Wasserstein Generative Adversarial Network, Vision Transformer, and Attack Discovery.

Introduction

The fourth revolution of Industry 4.0 (I4.0) with smart manufacturing factories enables novel opportunities and disruptive innovations by enabling the smart Industrial Internet of Things (IIoT) environment. The I4.0 environment comprises sensors, actuators, and smart communication elements. The IIoT devices are tiny in size and limited in resources like energy, bandwidth, and memory. Such devices monitor, collect, and transmit data through internet-based connections, creating a smart environment. Thus, the IIoT improves manufacturing efficiency by offering new opportunities and business models. The intelligence system automation of I4.0 is enhanced through smart machinery connectivity, resulting in tailored products and services [1]. Five primary stages are exploited to compose the I4.0-based smart manufacturing industries: sensing devices-based environmental monitoring, data collection, data transmission, data analysis, and data aggregation [2]. The various sensors, actuators, and intelligent devices initially monitor the surrounding environment and collect industrial information. Further, the data is collected by a high-capacity device and transmitted to the server location for analysis and aggregation.

Hence, different types of routing protocols are used for communication, and the Routing Protocol for Low-power and Lossy networks (RPL) is the familiar protocol that supports IPV6 communication over an IIoT environment [3]. Albeit the RPL-IIoT maximizes smart communication efficiency and improves productivity in manufacturing smart factories by marrying the digital

world, RPL security is a major challenge in such an environment. The ingenious hacking activities lead to different vulnerabilities and system damages [4]. Therefore, security in RPL- IIoT is a critical element, and it is crucial to focus on diverse security like known, unknown, and zero-day vulnerabilities in smartly connected digital devices of smart manufacturing factories. Generally, the RPL creates low powered DODAG structure among the devices and enables communications in the network. However, the technological diversity and resource limitation characteristics increase the RPL security challenges in vulnerability analysis and detection [5] [6]. Several security vulnerabilities, such as Denial of Service (DoS), Distributed DoS, rank, version number, Objective Function (OF), and zero-day, are present in the I4.0 environment [7].

A good security model should comprise two methods: vulnerability analysis and attack discovery. Firstly, the cyber security vulnerability analysis model analyzes the different types of known and unknown vulnerabilities using various models [8]. Fuzzing is the widely employed technique for vulnerability analysis in IoT systems [9]. Several fuzzing methods and tools are available for the current IoT environment. However, due to the massive heterogeneous data generation nature, they are not highly fit for the IIoT environment, especially in smart factory monitoring and control. Among the available fuzzing methods, the learning-based fuzzing models can increase system automation with high vulnerability distribution and minimum cost values compared to general fuzzing models. The Wasserstein GAN-based fuzzing can increase the vulnerability distributions even if the input data is highly dimensional. However, integrating fuzzing and learning models over the IIoT environment is a major question and thus increases the RPL security vulnerability analysis challenges [10]. The deep learning-based vulnerability models achieve high accuracy by automatically learning the massive IIoT data [11]. However, a lack of RPL-IIoT feature-rich datasets restricts the deep learning model performance, and they lack the ability to obtain complex patterns over the high-dimensional dataset. Image-based vulnerability detection with CNN plays a significant role in solving the abovementioned issues. Although CNN is considered the fundamental component to identify the attacks from images in industry 4.0 applications, the Vision Transformer (ViT) has become a potential alternative to CNN by effectively capturing the global relationships with attention mechanisms and improving the detection performance. Hence, the concept of a ViT with the advanced model is exploited to effectively capture complex spatial patterns or global relationships among the images. This solution remarkably increases the vulnerability and attack detection accuracy under the IIoT environment.

This paper proposes a Novel RPL Security (NRS) model with fuzzing for vulnerability and a vision transformer for attack discovery over RPL-enabled smart factory monitoring and control. The major contributions of the proposed model are as follows.

The primary objective of NRS is to analyze and discover the multiple types of vulnerabilities, including zero-day over RPL- IIoT. To perform vulnerability analysis and detect diverse attacks, the NRS integrates two methods: Wasserstein Generative Adversarial Networks (WGAN)-Fuzzing-based Vulnerability Analysis (WFVA) and Vision Transformer-based Attack Discovery (VTAD).

The proposed model collects the IIoT sample with routing-rich features using the Cooja simulator to construct the novel RID and improve the vulnerability distributions in fuzzing output data. By assuming that the novel zero-day vulnerabilities are partially similar to the existing vulnerabilities, the RID creates samples for zero-day vulnerability analysis and detection.

To analyze multiple RPl-IIoT vulnerabilities and to accomplish high distributions of vulnerabilities in fuzzy outputs, the WFVA creates multiple numbers of fake samples with the assistance of an automatic WGAN model. By executing the deep learning algorithm only at edge devices, the NRS effectively handles the resource limitation issues of IIoT with minimum cost.

The NRS boosts vulnerability detection accuracy by providing the high-vulnerability distributed fuzzer output converted into image patches as input to the VTAD model. Advanced attack detection strategy, a ViT in attack detection decision-making of NRS, significantly enhances accuracy.

Finally, the Python-based simulation results show the superiority of the proposed NRS. The results demonstrate the advantage of the proposed NRS in terms of various metrics: Fuzzer Output Recognition Rate (FORR), Diversity of Generated Data (DGD), accuracy, precision, recall, and F1-Score.

Literature Survey

This section divides the literature review into vulnerability analysis and discovery-based methods.

Vulnerability Analysis

The work in [12] analyzes the vulnerabilities related to power systems. It utilizes a Random Chemistry algorithm and a comprehensive Complete Cascading Failure Graph (CCFG) for vulnerability analysis. The study assesses vulnerability using CCFG-based indices, revealing variations under uncertainty. To mitigate the impact of uncertainty on system vulnerability, the vulnerability analysis model tests the system's resilience. Further, it analyzes the system risk level using different scenarios and utilizes a threshold value to measure the uncertainty level of the system. A cascading failure simulation (CFS) strategy in [13] analyzes the cascading failure propagation issues over integrated power-gas systems (IPGSs). It integrates different factors like generator and gas well ramping, transmission lines tripping and gas pipelines to manage the island issues and load-shedding problems. It exploits a hybrid learning model with a random forest and regression algorithm. Further, it classifies the vulnerabilities under different categories. A framework in [14] proposes an automatic vulnerability discovery with a hybrid neural network structure. It utilizes a program source code with fine granularity for inter-procedural vulnerability detection. It transforms the inputs by utilizing a lower-level virtual machine intermediate representation (LLVMIR) and backward program slicing model to enable analysis over control and data level. An integrated data mining framework in [15] proposes an automated vulnerability analysis model. It considers the vulnerability probability distribution model with Topically Supervised Evolution Model (TSEM) to analyze the distributions of vulnerabilities.

The research work in [16] introduces a vulnerability analysis model for scanning large-scale source code. The main intention of such work is to improve scalability and accuracy. Further, it includes a deep learning strategy to classify images. Another work [17] proposes an innovative deep-learning model called LineVD to analyze the vulnerabilities in the statement level of source code [17]. It designs vulnerability detection as a node categorization model and captures the dependencies among data and control packets based on graph neural. Also, it encodes the tokens of raw source code using a transformer-based model. By effectively handling conflicting outputs between function-level and statement-level information, LineVD significantly increases the performance of the prediction level, even if the function code vulnerability status is unavailable. Moreover, it utilizes a pioneering strategy to jointly learn the information from function and statement levels with the assistance of neural networks, resulting in accurate vulnerability analysis.

A deep learning-based penetration testing framework, LSTM-EVI, has been proposed in [18]. It utilizes a smart airport-level testbed model to detect vulnerabilities. It collects realistic IoT information by integrating physical IoT devices with virtual elements. Capturing benign and malicious or scanning data within the smart airport test bed increases vulnerability detection efficiency. The work in [19] aims to detect the OF-based vulnerabilities in the RPL-IoT environment using machine learning. It determines the combined attacks against two objective functions: Minimum rank with hysteresis objective function (MRHOF) and objective function zero (OF0). It constructs a novel IoT data set for efficient vulnerability analysis. Further, it utilizes cutting-edge technology to improve the performance level of the vulnerability analysis model. The work in [20] proposed an iDetect vulnerability analysis model that detects vulnerabilities in the C/C++ source code of embedded IoT operating systems. It exploits machine learning tools for vulnerability identification. The iDetect design consumes minimum energy and is highly suitable for detecting vulnerabilities in IoT environments. The work in [21] includes the widely used IIoT Computational Intelligence Algorithm (CIA) to analyze the vulnerabilities in a dark web environment. The demonstrates real-world hidden security vulnerabilities by creating a scenario. However, the above-discussed vulnerability analysis model fails to detect the RPL-I4.0 vulnerabilities, increasing the cost and complexity in IoT scenarios.

Vulnerability and Attack Detection Using Fuzzing Method

The fuzzing-based vulnerability detection models can improve the system automata with minimum costs. EOSFuzzer [22] is an advanced black-box fuzzing technique that identifies vulnerabilities in EOSIO smart contracts. It exploits effective attacking scenarios to test the oracles. It consists of various components: input generator, executor, instrumented Wasm VM,

and vulnerability detection engine for efficient vulnerability detection. The work in [23] presents an EtherFuzz model, a mutation fuzzing strategy that aims to identify Transaction Ordering Dependent (TOD) vulnerabilities over smart contracts. It precisely detects the vulnerabilities using a defined test oracle with mutation amplification test data. However, the unique operating environment and complex program characteristics make the Ether fuzzer inefficient for resource-constrained environments. The work in [24] presents a GAXSS approach that efficiently determines cross-site scripting vulnerabilities in web applications. It addresses vulnerability detection using a novel genetic algorithm-based fuzzing model. It improves the attack detection efficiency with minimum detection time by increasing the system automata and reducing algorithm structure complexity. A cutting-edge model called HotFuzz has been proposed in [25]. It detects the Algorithmic Complexity (AC) vulnerabilities automatically in Java libraries. It also integrates a genetic algorithm for vulnerability detection. The work in [26] presents a novel fuzzing model, named BECFuzz [26], to effectively address the following key challenges of the fuzzing model: collision at the edge, source code availability, and efficiency. The work in [27] proposes a fuzzing-based MQTT cybersecurity model for vulnerability detection in IoT networks. It inputs the ransom data to the fuzzing model and can detect novel security breaches in MQTT through effective vulnerability analysis. It increases the system automata and minimizes the computation cost in a resource-limited IoT environment.

The work in [28] presents a MultiFuzz model, a coverage-based multi-party protocol strategy utilized to detect vulnerabilities in IoT publisher/subscriber protocols. It provides a single input with multiple connections to increase the fuzzer output efficiency. It stimulates the transitions of publisher/subscriber by integrating a mutation algorithm. Moreover, it feeds the fuzzer outputs for testing by employing a multi-model. The work in [29] presents a protocol fuzzing technique to detect the unknown software vulnerabilities of industrial network protocols. Initially, it clearly understands the protocol fuzzing concepts to utilize it for vulnerability detection. Further, it applies the fuzzing-based detection model over RabitMQ with the assistance of the MAVlink protocol process. Thus, it improves the security in drone-to-ground-based communication models of industrial networks.

The work in [30] presents a new test technique with a protocol fuzzing model to detect zero-day security vulnerabilities in IIoT environments. It utilizes a black box testing model in which the test cases are generated using seed pools. It exploits diverse program paths to update the seed pool model-based test cases. Further, it utilizes three steps that are input, output, and delta, to search the novel program areas over a black box environment and detect the zero-day vulnerabilities. The main intention of the vulnerability analysis system in [31] is to select the best communication protocols for IoT and Web of Things (WoT) environments. It utilizes three intelligent objects to choose messaging and application protocols in non-critical and critical multi-level IoT and WoT. It determines the three objects by initial data generation, effective vulnerability analysis, and vulnerability discovery models.

However, the data generated by the IIoT environment is captured in terms of tabular form, in which the performance is hindered when the dataset comprises numerous massive real-time attributes. The existing learning models with numerical datasets fail to capture the spatial relationships among the significant features, resulting in poor detection performance. Converting the tabular data into images is a significant solution to the abovementioned issue [32]. Another work in [33] proposes a novel solution in which CNN is exploited to identify heart failure patients' survival status accurately. The CNN-based novel model converts the numeric data to the image to improve the learning efficiency and reduce detection errors. Generally, CNN plays a significant role in image-based attack detection. However, it fails to obtain the global relationships precisely. Therefore, the ViT-based strategies receive high attention that can precisely capture the complex patterns and improve the detection accuracy.

The work in [34] includes a ViT-based strategy to detect lung diseases effectively and classify them under different categories. It classifies the X-rays under various classes by including an off-the-shelf ViT model. The comparison results demonstrate that the ViT with a CNN-based hybrid deep learning-based detection model outperforms conventional deep learning methods regarding attack detection and efficiency. The work in [35] uses the ViT to classify the images in remote areas. However, there is no effective solution to convert the RPL-specific numeric dataset into images for improving the RPL-based industry 4.0 vulnerability detection. In addition to that, the existing model uses

fuzzing strategies for vulnerability attack analysis, and they do not attain better results when applying the RPL-I4.0 environment. Also, they need high manual analysis and minimize the vulnerability distributions in output owing to the errors that happened by manual analysis. Therefore, innovating a novel IIoT RPL security model is crucial that effectively learn the realistic massive features with fine-grained spatial relationships to improve vulnerability detection efficiency in the RPL-IIoT environment.

Problem Statement

In the current world, IIoT exploits different industry 4.0 standards to connect different systems that interact with the physical world. The diversity in 4.0 standards with various network connections increases the vulnerabilities, especially in RPL-based routing services. Hence, efficient security solutions are needed to provide defense against such vulnerabilities. A lot of vulnerability analysis and attack discovery models are presented in the literature. However, they fail to fulfill the RPL-IIoT environment's security issues effectively. Firstly, the existing works mainly focus on devices and application layer vulnerabilities. Hence, there is no unique security model for the RPL-based IIoT environment. Secondly, the benchmarking datasets do not contain RPL-rich features and data. Thus, it limits the evaluation efficiency of RPL-IIoT. Hence, there is a need to design an RPL-IIoT dataset for effective evaluation. Thirdly, the general fuzzer and other fuzzing models increase the manual analysis and decrease vulnerability data distribution in fuzzy outputs. Many errors happened due to protocol specifications and misunderstandings. Finally, the conventional deep learning-based vulnerability discovery models inherently pose a higher degree of spatial variance and improve the false alarm rate due to ineffective learning with non-fuzzing inputs. It improves the accuracy with a minimum level when adopting fuzzing input datasets as images. A ViT is an advanced learning strategy that obtains fine-grained spatial relationships that surpass the CNN, thereby enhancing vulnerability discovery accuracy over RPL-enabled industrial 4.0. Moreover, it is crucial to avert such issues by designing a novel RPL-IIoT security model with fuzzing and ViT models.

Network Architecture and Vulnerabilities

A set of tools and applications define the IIoT in which large enterprises accomplish end-to-end connectivity from core to edge. Industrial 4.0 covers smart manufacturing factories, digitalization, and corporate sustainability in its broader future scope. The applicability of RPL routing in such a smart factory monitoring and control environment needs significant attention. The RPL routing is applied among the smart factory sensors, devices, and edges. The smart factory monitoring environment may introduce different challenges: signal interference and temporary disruptions in communication. Applying RPL routing in such an environment is highly resilient to packet loss and can easily adapt to lossy network conditions. Thus, it assures reliable communication even if the network comprises subtle changing conditions. Also, the hierarchical structure in the RPL protocol can effectively organize the smart factory monitoring devices and enable optimized routing paths for communication. In the proposed NRS, the RPL can be configured to provide well-suited support to real-time communication requirements, and it facilitates timely routing services to the monitoring data. It is crucial for applications with rapid responses, such as in-process monitoring and control for equipment over smart factory I4.0. The proposed smart factory monitoring I4.0 architecture comprises five different layers: cloud layer, edge layer, IoT gateway layer, network layer, and physical resource layer in which everything is interconnected using 5G internet completely, and they work automatically. Figure 1 shows the architecture of the I4.0 environment.

Cloud Layer: This layer comprises various servers, cloud, data, and application servers to perform in-depth analysis, offer high storage, and handle resource limitation issues. The devices or users can obtain information from cloud servers anytime and anywhere.

Edge Layer: The edge layer comprises different edge devices that take the cloud services near the users or devices. It also manages the resource limitation problems of I4.0 devices by executing complex algorithms instead of those at resource-limited devices.

IoT Gateway Layer: The sensor data is collected and converted into digital channels to process further at the internet gateway. Further, it transmits the collected digital data through the internet for further processing before transmitting it to the cloud. The gateways are part of the data-collecting systems of edges. The gateways are adjacent to the physical layer and preliminary to the edge layer.

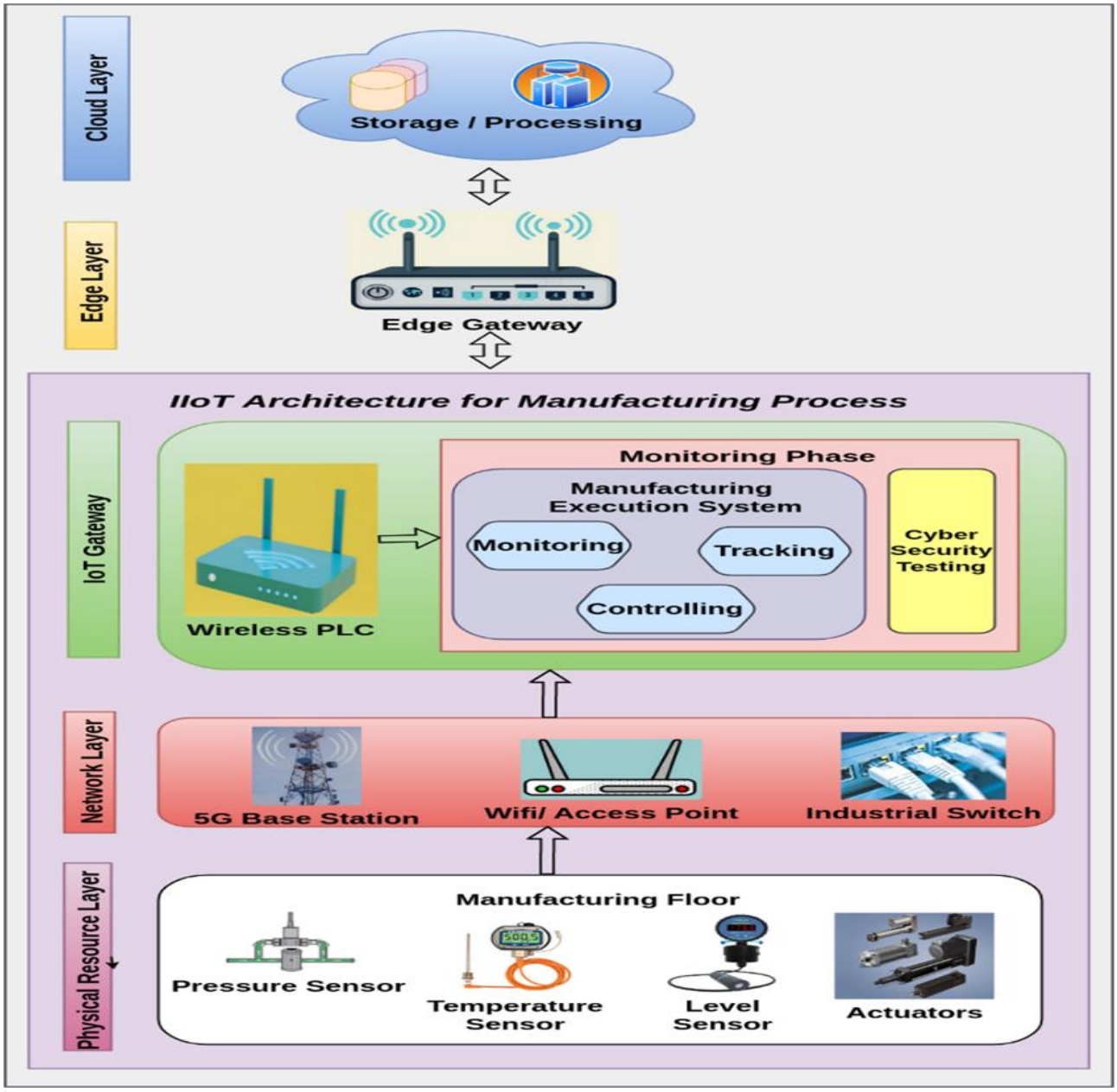


Figure 1: Architecture of Smart Manufacturing and Control of I4.0

Network Layer: It consists of 5G base stations, wireless access points, and industrial switches to enable data communication. The data of the IIoT system are transferred by this layer using various communication protocols. Such protocols should be designed based on industry standards and requirements. The protocols define the law for data transfer among diverse IIoT devices and comprise the security features for reliable communications.

Physical Resource Layer: This layer comprises various sensors such as pressure, temperature, and actuators to continuously monitor and provide timely reports over the smart factory environment. The sensors are tiny in size and limited in resources like energy, memory, and computation. The sensors monitor the specific environment and obtain the data. Further, it collects and sends the information to the server location for further action.

IIoT Network Vulnerabilities

The RPL is the most widely used protocol for IIoT. Albeit the RPL provides reliable routing services in a resource-constrained environment, it lacks strong security features against different vulnerabilities. It does not incorporate strong security against the vulnerabilities of the industrial 4.0 IIoT environment. RPL is exposed to

several vulnerabilities, such as objective function, zero-day, denial of service, and rank attacks.

OF Vulnerability: The RPL exploits two familiar OFs that are OF0 and MRHOF. The OF plays a significant role in DODAG construction and routing processes. There is a possibility for OF vulnerabilities, like version number, rank, and parent selection, in RPL-IIoT. Considering the vulnerabilities related to OF is crucial as it significantly impacts the network performance.

Zero-Day: It is a serious type of vulnerability, as the manufacturers do not know the vulnerability pattern of such attacks. This kind of new attack/hack was previously unheard of in the community as a zero-day vulnerability. Zero-day vulnerabilities are considered a potential security risk, and it is not easy to design defenses against them as they have not happened in the network previously. The proposed NRS takes assumptions on the zero-day sample creation that it is partially similar to the existing vulnerabilities to model it in the smart factory environment effectively.

Rank: In this type, a malicious device intends to announce a wrong rank, leading the neighboring devices to select the malicious device as a preferred parent. It is very dangerous in an RPL-based IIoT environment, as the wrong parent selection can shrink the entire RPL performance.

Version: In this type, the malicious node manipulates or exploits the version number field of the RPL protocol to launch the attacks. This type of attack is highly related to DoS, which is inaugurated by increasing the RPL control traffic during the global repair mechanism.

Theft: In this type, the attacker may try to manipulate the RPL routing table maintained by entities. This way, the attacker can redirect the traffic, isolate the nodes from routing participation, and disrupt the normal routing function.

Blackhole: In this type, an adversary selects the blackhole nodes to discard or drop the routing packets, thereby minimizing the RPL performance.

Sinkhole: In this type, an adversary can divert communication traffic toward a compromised or malicious node, permitting the attacker to eavesdropping or data tampering.

Selective Forwarding (SF): In this type, a malicious node selectively drops certain packets or refuses the packets while allowing other nodes to pass through. In this way, it can disrupt normal RPL operations.

Hello Flood (HF): An adversary aims to flood a significant or large amount of malicious control packets and disrupts the normal routing function of the RPL protocol. This attack leads to high energy exhaustion, poor network performance, and increased traffic.

Distributed DoS Vulnerability: It is a type of DoS attack where the incoming traffic originates from multiple distinct sources, and it is challenging to restrict such an attack by blocking a single traffic source. Such attacks are analogous to the gate of a shop being blocked by a crowd of non-customers, thus disrupting trade.

Overview of the Proposed Model

The revolutions in I4.0 advocate the novel technologies and automation of conventional industrial structures and manufacturing. However, security is a great concern that can impact the surveillance of I4.0 systems. This paper proposes a novel IIoT security model, NRS, that integrates WFVA and VTAD strategies. An overview of the proposed methodology is explained in Figure 2.

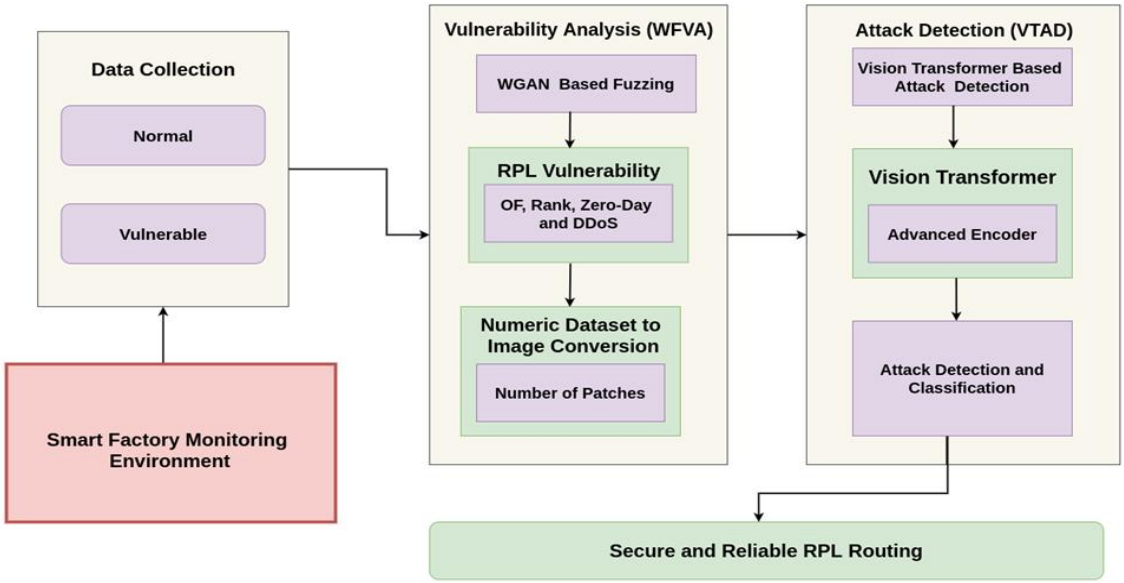


Figure 2: Overview of Proposed Work

The proposed model comprises two main parts: RPL vulnerability analysis and attack discovery. It utilizes the WGAN fuzzing model to generate the input fuzzy sets and to analyze the RPL vulnerabilities. Fuzzing is an effective method to discover vulnerabilities in industrial RPL protocol. The WGAN-based fuzzing is highly adaptable, as it reduces the manual analysis of other fuzzing techniques and effectively deals with massive input data over an IIoT environment. Initially, the WGAN-based vulnerability analysis model collects information from the smart factory monitoring environment of I4.0. The fuzzing model is used to generate the fuzzing data for vulnerability analysis. Consequently, it converts the fuzzing data into various image patches and feeds the image dataset as the input to the ViT-based attack discovery model. The second part of the proposed work utilizes a ViT that exploits an advanced transformer encoder to effectively observe the environment's complex vulnerability patterns and detect various RPL vulnerabilities from the image-based fuzzing dataset. Moreover, the proposed work can effectively analyze and discover attacks of RPL-based IIoT with high automation using suitable fuzzing and ViT strategies.

WGAN-Fuzzing for Vulnerability Analysis (WFVA)

The WGAN is a variant of the GAN model that can highly scale with large data sets and is stable with large-scale I4.0 manufacturing applications. Compared with other fuzzing techniques, the learning-based fuzzing model minimizes manual analysis and automates the system without sacrificing performance. Therefore, the NRS incorporates the WGAN-Fuzzing for vulnerability analysis. Exploiting heterogeneous technologies, highly-structured data generation, prone to diverse novel attacks, and resource-limited tiny industrial device characteristics escalate the security challenges in the smart factory monitoring IIoT environment. By integrating the WGAN-Fuzzing model to analyze the RPL vulnerabilities over IIoT and to generate the fuzzer outputs with recent and zero-day vulnerabilities, the WFVA increases the attack detection accuracy and improves the RPL routing performance over resource-limited IIoT devices. The NRS executes the WGAN-Fuzzing only at edge IIoT devices to manage the resource limitation issues associated with IIoT devices effectively. Finally, the fuzzing output dataset is converted into an image dataset, making it highly suitable for ViT-based attack detection. The WFVA process is explained in the following Figure 3.

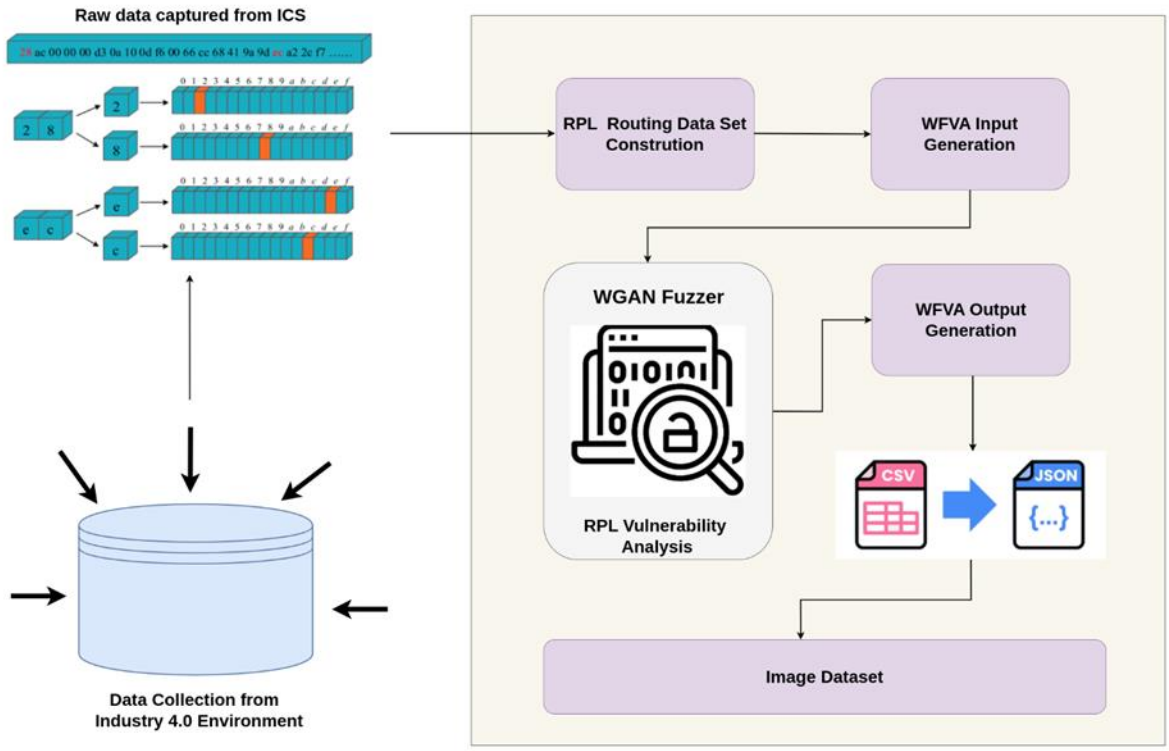


Figure 3: Process of WFVA

Data Collection of IIoT and RID Construction

Initially, the WFVA collects the RPL routing data from the industrial 4.0 manufacturing environment to construct the RPL-IIoT Dataset (RID). The proposed NRS model generates the RID using the packet capture (PCAP) files from Contiki-based IIoT devices in a virtual industrial 4.0 network simulation platform utilizing the Cooja simulator. The NRS constructs the novel RID from a diverse IPV6-based network environment, which is frequently utilized in large-scale IIoT deployment scenarios and different types of low-power IIoT devices are supported to construct such networks. Further, the RID dataset will be evaluated with 100 industrial IoT nodes, and hence, it can be exploited to design a scalable security vulnerability analysis and attack discovery model for an IIoT network. The RID dataset is a good solution for RPL-IIoT security design as it has lightweight network packet traces which can be employed over resource-limited IIoT devices. The constructed RID comprises different network vulnerability types: OF, zero-day, rank, and DDoS,

Data Sample Creation: The proposed NRS exploits the Cooja simulator to collect the data from the IIoT environment and construct the RID with a rich set of vulnerabilities. The NRS generates the IIoT data from the industrial 4.0 environment using Cooja. Thus, it effectively handles the lack of IIoT vulnerability datasets. Initially, by running the RPL-IIoT code in Cooja, the simulator collects the RPL traces as raw packet capture (PCAP) files from the environment. After data collection, the data normalization is applied to the raw samples. An example of raw data collection samples is depicted in Table 1.

Table 1: A sample of Raw RPL-IIoT Dataset

No	Time	Source	Destination	Packet Length	Information
121	8.**84569**	SID12::c30c::0::2	DID12::c30c::0::3	102	RPL Control Message
122	8.84784	SID12::c30c::0::6	DID12::c30c::0::8	97	RPL Control Message
123	8.841086	SID12::c30c::0::12	DID12::c30c::0::18	76	RPL Control Message

Data Normalization: Generally, the I4.0 manufacturing scenario is very dynamic, and the data collected from such an environment have varying central tendencies. Hence, the NRS exploits the min–max normalization technique to normalize the attribute values into a common range. The equation for the min-max normalization value of RID ($\mathrm{RID_{Norm}}$) is given as follows.

$$\mathrm{RID_{Norm}} = \frac{\mathrm{RID_{act}} - \mathrm{RID_{min}}}{\mathrm{RID_{max}} - \mathrm{RID_{min}}} \qquad (1)$$

In equation (1), the term $\mathrm{RID_{act}}$ denotes the actual value. The terms $\mathrm{RID_{max}}$ and $\mathrm{RID_{min}}$ refers to the maximum and minimum values for normalization, which is decided using actual value. Further, the proposed NRS utilized the WGAN-Fuzzing to analyze the vulnerabilities of RID and generate the WFVA output generation. The WFVA output comprises ten types of vulnerabilities: OF, zero-day, rank, DDoS, version, theft, blackhole, sinkhole, SF, and HF.

PCAP to CSV Transformation: The NRS transforms the PACP files of the normalized dataset into Comma Separated Values (CSV) with the assistance of Python libraries. Generally, the collected PCAP files are huge, and the NRS dissects the PCAP into CSVs using Wireshark. Further, the converted CSV files are aggregated and sent as input to the WGAN-Fuzzing Model. The sample dataset is given in the following Table 2.

Table 2: Dataset Details

Dataset	Benign			Vulnerabilities
	Scenario	Devices	Total Packets	Total Packets
OF	OF10	10	1,60,820	8,415
Zero-Day	ZD10	10	0	0
Rank	R10	10	1,60,820	5,175
DDoS	DDoS10	10	1,60,820	3,857
Version	V10	10	1,48,760	4,985
Theft	T10	10	95,760	2,684
Blackhole	B10	10	1,34,600	5,288
Sinkhole	S10	10	1,38,778	6,825
SF	SF10	10	1,05,610	5,885
HF	HF10	10	1,32,157	5,175

Further, the preprocessing steps are applied over the datasets only to extract the RPL-rich features for vulnerability analysis and attack detection. After feature extraction, the dataset comprises the features described in

Table 3.

No	Features	Description
1	No	Sequence number of network packet
2	SID	Source device Identity
3	DID	Destination device Identity
4	L	Length of the packet
5	PTR	Packet Transmission Rate
6	PRR	Packet Reception Rate
7	ATT	Average Time for Transmission
8	ATR	Average Time for Reception
9	CTP	Count of Transmitted Packets
10	CRP	Count of Received Packets
11	minute	Minutes (from Time)
12	second	Seconds (from Time)
13	millsec	Milliseconds (from Time)
14	Label	Types of Attack
15	RPL_DAO	RPL control packets
16	RPL_DIO	RPL control packets
17	RPL_DIS	RPL control packets

Table 3: Extracted Features of RID

The RID network packet traces with extracted features are provided as input to the WGAN fuzzing model. The WGAN-Fuzzing starts to generate the output test cases of fuzzer outputs by taking the RPL-IIoT network packet trace as inputs. The process of WGAN-Fuzzing is explained in algorithm 1.

```
Input: Raw Dataset
Output: WGAN-Fuzzing Output Dataset
Initializes the WGAN Fuzzing model;
        Feeds the raw dataset as input to the WGAN-Fuzzing model;
WGAN do {
        For (Attack=n; n≤10; n++) {
                Generate the initial test case for attack n;
                Executes the fuzz test with real data in the dataset;
                Identify the targeted fuzzed outputs through analysis;
                Final test case generation;
                }
        Consolidates the n test cases;
        Constructs the WGAN-Fuzzer output dataset;
}
```

Algorithm 1: WGAN-Fuzzing

Vulnerability Analysis over RPL-IIoT Protocol

Owing to the resource-constrained nature of IIoT devices and the unsecured internet connectivity of the RPL routing protocol, RPL-IIoT is prone to various attacks. The RPL-IIoT routing is prone to various attacks that target different functionalities of the DODAGs like topology disruption, unwanted network resource utilization and traffic misrouting. Developing novel security solutions to RPL protocol and performing regular vulnerability assessments for auditing and addressing the potential vulnerabilities over RPL-industry 4.0 is crucial. Therefore, the NRS model incorporates WFVA to analyze such types of RPL vulnerabilities clearly and to generate vulnerability-based fuzzing output data. The fuzzing models are generally used to analyze or detect network vulnerabilities. It builds the test cases according to the inconsistent protocol specifications in many scenarios. The disregarded information of RPL protocols is vulnerable to zero-day or novel attacks. Also, the conventional fuzzing techniques need a high cost to implement in the I4.0 scenario. The WGANs are also called deep adversarial learning models, which do not require protocol specifications and are not too expensive to generate the fuzzing output. Therefore, the NRS exploits the more stable and higher scalable WGAN-fuzzing model for RPL-IIoT vulnerability analysis.

WGAN-Fuzzing Model Design: The WFVA intends to intelligently analyze the vulnerabilities from raw network packets of RPL-IIoT from RID. After vulnerability analysis, it can obtain a concrete fuzzy output data generation model to generate well-formed learning data for attack discovery. Thus, it provides accurate learning sets to the learning model and increases detection accuracy. The WGAN-Fuzzing consists of two models: are generator and a discriminator. The following Figure 4 shows the architecture of the WGAN-Fuzzing model.

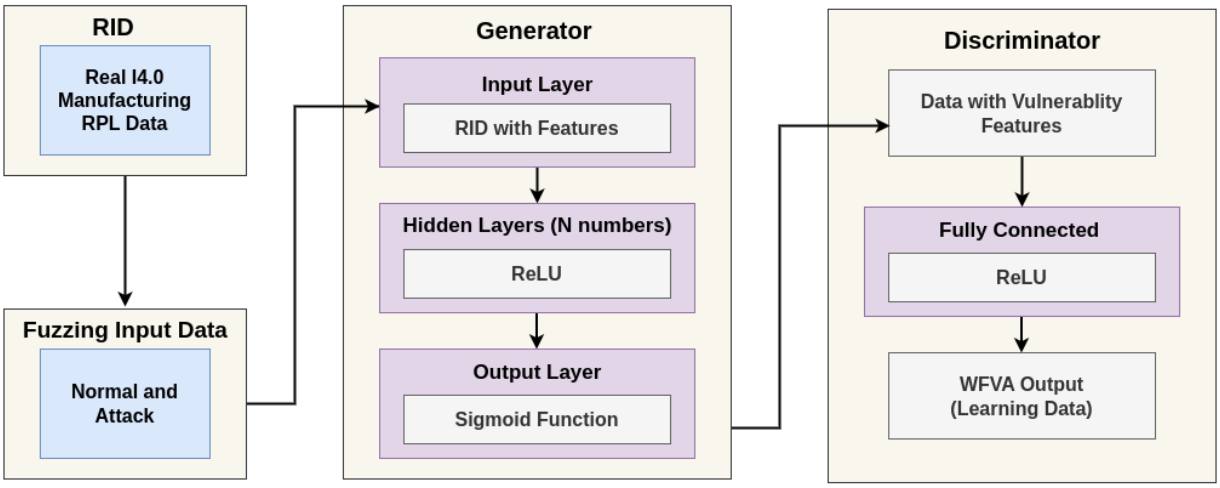


Figure 4: Design Model of WFVA

Firstly, the generator model comprises three layers: input, hidden and output. The proposed WFVA adjusts the number of hidden layers according to the size of the RID. Regarding the discriminator of the GAN network structure, the WFVA utilizes the fully connected neural network structure as an activation function in the input layer of WGAN-Fuzzing. Thus, it reduces the calculations on the generator side. The maximum and minimum lengths of IIoT features are set using the input data size of the neural network or WGAN. Thus, it processes the data uniformly. The WFVA utilizes the rectified linear units (ReLU) as an activation function in the middle layer, as it neglects the vanishing gradient problem in training and averts the overfitting issues caused by zero-valued neurons with minimum computational costs. Finally, the WFVA selects the sigmoid function as an activation function in the output layer of WGAN-fuzzing. The output values lie between 0 to 1. The WGAN employs the Earth Mover's Distance (EMD) to compute the difference between two feature distributions. It is estimated using the following equation (2).

$$W(f_i, f_j) = \inf_{k \in \prod(f_i, f_j)} E_{(i,j)\sim k}\left[||i - j||\right] \quad (2)$$

In equation (2), the terms $f_i \; and \; f_j$ represent the two feature distributions. The term k is the joint distribution value of $f_i and \; f_j$. The generator estimates the joint data values for each feature and provides it as input to the discriminator model. Consequently, the discriminator generates the output fuzzer values to the corresponding input values.

Fuzzing can analyze and discover security vulnerabilities in a protocol by providing a large amount of unexpected input to the learning model. A significant part of WFVA is output data generation. The proposed NRS model considers four types of RPL vulnerabilities. Hence, different feature sets are essential to analyze such types of attacks. The key features crucial to discovering such types of attacks are discussed as follows. Firstly, the objective functions and rank fields are generally used to analyze the OF and rank vulnerabilities. Secondly, the number of packets and energy fields are used to identify the DDoS attack. However, the zero-day attacks are unknown, as they have not happened till now in the network. The WGAN-Fuzzing model provides random and invalid input sets to the GAN model to generate future zero-day attack samples. Further, it is used to train the learning algorithm for vulnerability discovery. Algorithm 2 explains the WFVA model.

//WFVA Model//

Input: Real-time RPL-I4.0 and RID

Output: Fuzzy Output Data with Vulnerabilities

NRS do {

Collects the realistic data from I4.0 environment;

Constructs the RID for vulnerability analysis and discovery;

Divides the RID into learning and testing;

Applies min−max normalization technique using equation (1);

Provides the learning data to the WFVA for analysis;

WFVA do {

Generates fake samples related to realistic data;

Compares the fake sample with multiple hidden layers;

Hidden layer (n=Size (RID); n++);

Generate output fuzzer values with data diversity;

}

Feeds the output fuzzer values as input to VTAD for attack discovery;

}**};**

WFVA Output Data Generation

Algorithm 2: Working Process of WFVA Model

Fuzzing Data to Image Conversion: The fuzzing output data is in CSV file format. Hence, converting the CSV files as images to input the fuzzing output data for the ViT-based vulnerability discovery model is crucial. This process is shown in the following figure 5.

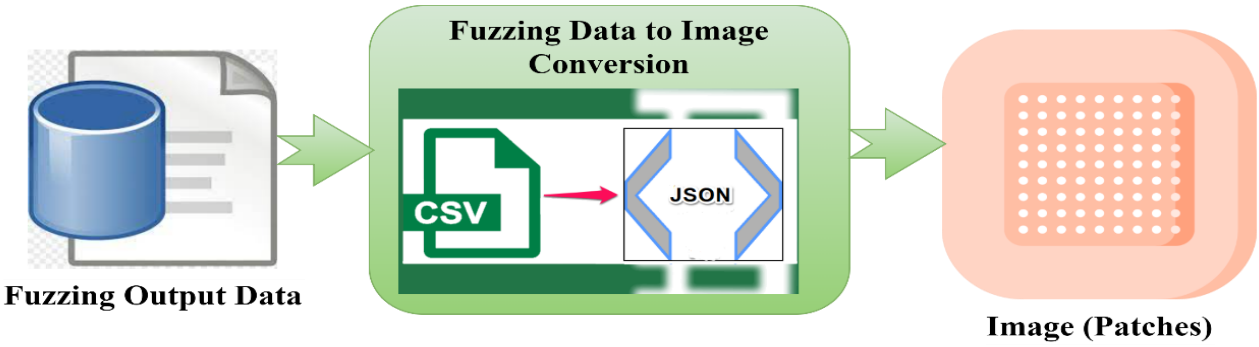


Figure5: Fuzzing Output to Image Conversion

The new CSV files of fuzzer output are in different sizes in terms of number of rows and columns. Therefore, the proposed NRS applies some preprocessing steps to ensure that the images have the right sizes that are highly fit for the modules in the ViT. The mapping step is used to convert the CSV as JSON image values. Firstly, it initializes a blank image with the same height and width as the number of rows and columns of the fuzzer output CSV file. Consequently, each feature value from the fuzzer output CSV file will be converted into a JSON value with the appropriate color. It is accomplished by determining the highest and lowest value of each feature in the fuzzer output dataset and then exploiting this feature range to map the feature value linearly to the JSON range.

Generating RPL-IIoT data benefits precise vulnerability detection greatly, and at the same time, it is a major disadvantage that numerical data with deep learning solutions significantly minimizes the vulnerability discovery rate. They fail to precisely capture the complex learning patterns of high-dimensional RPL datasets and diminish the detection performance. To rectify this issue, the NRS creates an image of each test case generated by the WGAN fuzzer. Further, the NRS maps the highly correlated RPL features for attack detection at nearby locations of the image and improves the image conversion quality of the RPL numeric dataset. Different image samples are created from the same sample during image conversion by performing rotation, feature location changes, and pixel variations. Thus, it increases the number of image samples in the training dataset and maximizes the performance of vulnerability detection. In the image, the NRS identifies the brightness of each region by representing feature significance values between 0 and 1 of the data. The images are created in gray space. Moreover, the image dataset is provided as input to the VTAD model for vulnerability detection.

Vision Transformer-based Attack Discovery (VTAD)

The proposed NRS model integrates a novel ViT strategy for vulnerability discovery. The ViT can obtain fine-grained spatial data from the environment and enhance image-based attack detection accuracy in an RPL-enabled smart monitoring industry 4.0 environment. Figure 6 shows the architecture of the ViT integrated in NRS. The NRS feeds the RID with the images as input to the ViT. The ViT has three layers: input, transformer encoder, and output. The input layer obtains the input images as different patches and sends them to the transformer encoder. The transformer encoder comprises two main parts: multi-head self-attention and feed-forward. In the self-attention layer, the input images are transformed into three vectors: query, key, and value with dimension. The ViT transformer derives from different inputs to pack them into three various matrices to construct and estimate the self-attention function. The vectors that have high probabilities receive more attention in the upcoming layers. The multi-head in the self-attention layer is a mechanism that can be exploited to boost the efficiency of the self-attention layer.

The transformer applies a feed-forward for linear transformations to each encode and decode process of the self-attention layer. The transformer output is provided as input to the output or SoftMax layer. In the proposed NRS, the number of layers is adjusted according to the dataset dimensionality, making the

NRS more suitable for RPL-industry 4.0 vulnerability detection. Finally, SoftMax decodes the transformer outputs and determines the vulnerabilities according to the self-attention significance. The ViT learns the input image dataset using multiple layers, and the generator generates a fake information value for feedback generation. Further, it matches the learning data with the testing data and determines the vulnerabilities through the encoding and decoding with multi-head self-attention. Finally, SoftMax generates the output to the corresponding input. At this stage, the attention to the NRS focuses on the last layer to obtain the best detection results. Finally, it combines the features identified in the previous phase as input to the classifier, which is performed according to the SoftMax layer. In this way, the NRS determines the vulnerabilities most accurately. Compared to the transformer activation function, the output of the SoftMax layer activation function is unique, as it plays a significant role in vulnerability discovery.

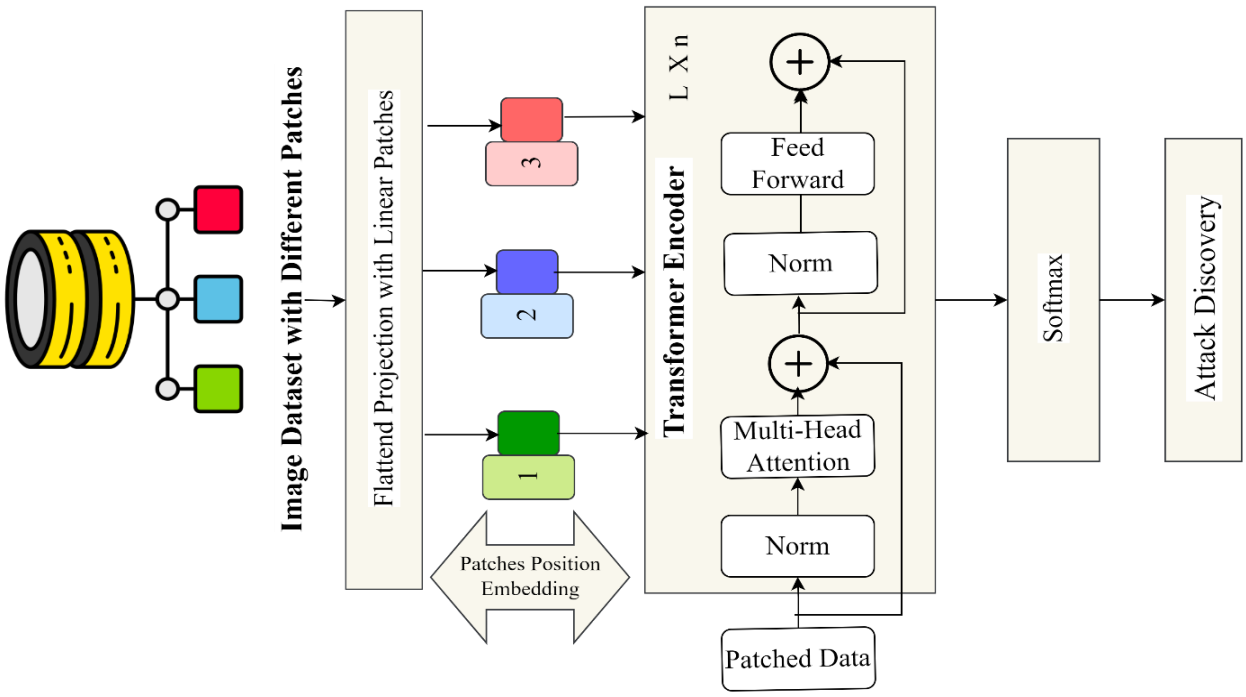


Figure 6: Architecture of Vision Transformer

Initially, the NRS provides the image output data as input to the VTAD model for attack discovery. The ViT performs learning with the output of the WFVA image dataset with labels. Further, they start the attack discovery on the testing data. Further, it compares the testing data with the learning information and provides the output with attacks. Moreover, the VTAD classifies the detected attacks under different classes like OF and rank, Zero-day, DDoS, version, theft, blackhole, sinkhole, SF, and HF. Algorithm 3 explains the VTAD model.

//VTAD Model//

Input: Image Dataset;

Output: Attack discovery under different classes;

ViT do {

Initiate the learning process with image dataset;

Transformer do {

Perform learning with labeled fuzzing output data;

Initiates the testing process for attack discovery;

Discover the attacks using different layers;

Classifies the attacks under different classes;

}};

Algorithm 3: Working Process of VTAD Model

Experimental Evaluation

The effectiveness of the proposed approach is analyzed using Python machine-learning libraries. A personal computer with Intel i5 2.5GHZ CPU and 16 GB memory is utilized to carry out the experiment. Using deep learning strategies, the NRS exploits RID to build the learning and testing phases. The Contiki is highly suitable for simulating the IIoT environment, as it is a reliable and widely used open-source network simulator that can enable connections among tiny, low-cost and low-power industrial devices. It also supports the RPL protocol for low-power IPv6 networking and the 6LowPAN adaptation layer protocol. For evaluation, the proposed WGAN-Fuzzing model is compared with the existing general fuzzer [36] and GAN-based fuzzer [37]. Further, the vulnerability detection efficiency of the ViT model is evaluated by comparing it with the existing deep learning algorithms. The following Table 4 shows the simulation parameters of the Cooja.

Table 4: Simulation Parameters for Dataset Generation

Parameter	Values
Simulator	Contiki Cooja
Protocol	RPL
Simulation Area	500m*500m
Number of Nodes	100
Number of Attackers	10
Physical Layer	IEEE 802.15.4
Radio Medium	UDGM
Transmission Range	50 m
Simulation Time	300 seconds

Performance Metrics

The results are obtained using the following metrics.

Vulnerability Analysis:

Fuzzer Output Recognition Rate (FORR): It is the percentage of output generated by the WGAN-Fuzzing Model.

Triggered Efficiency of Vulnerabilities (TEV): It denotes the number of vulnerabilities triggered by WGAN fuzzing after vulnerability analysis.

Diversity of Generated Data (DGD): It is the ability to maintain diverse learning data of the WGAN-Fuzzing model.

Attack Discovery:

Detection Accuracy: It is the percentage of vulnerabilities that are correctly identified as attacks.

Precision: It is the percentage of correctly identified vulnerabilities.

Recall: It is the percentage of correct positive predictions from the actual positive samples in the dataset.

F1-Score: It is the combination of precision and recall.

Experimental Results

Vulnerability Analysis Results:

To evaluate the effectiveness of the proposed WGAN-Fuzzing model, the learning epochs are varied from 10 to 100.

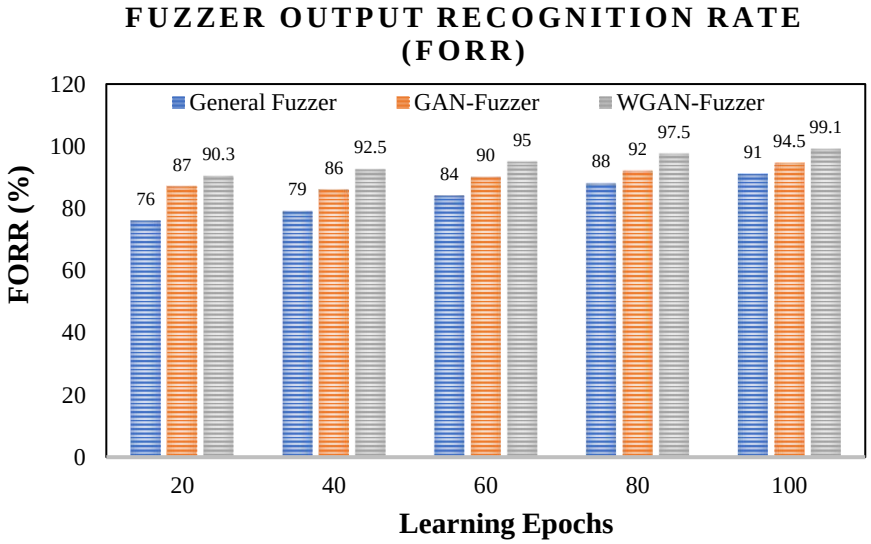


Figure 7: Learning Epochs VS FORR

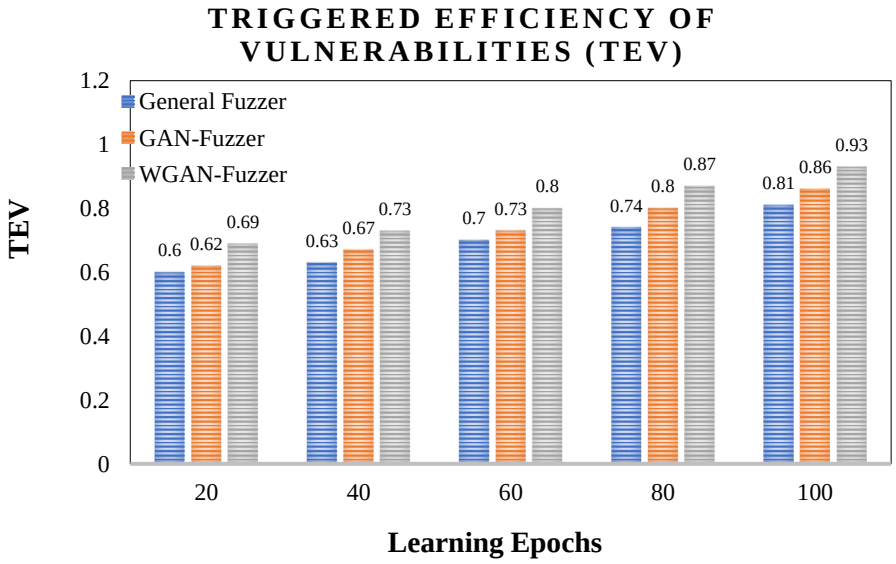


Figure 8: Learning Epochs VS TEV

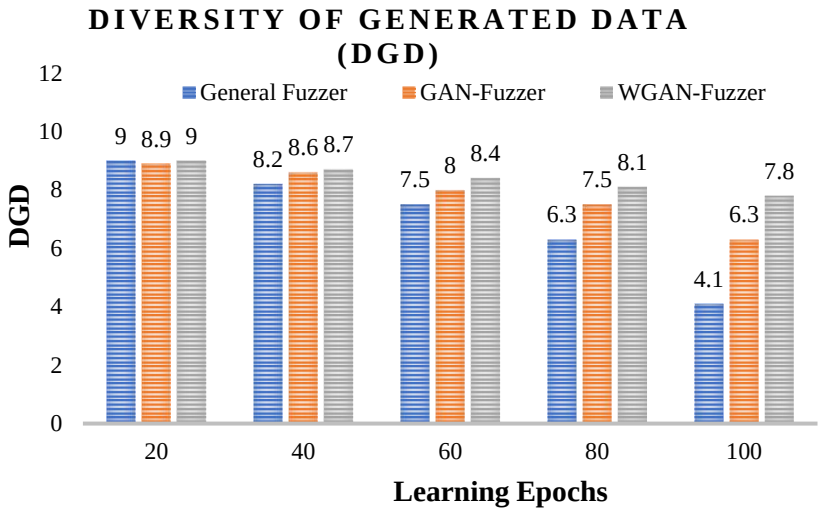


Figure 9: Learning Epochs VS DGD

The WGAN-Fuzzing-based vulnerability model simulation results are taken by providing 70% of the RID as fuzzing input data. The input samples are divided by various epochs to analyze the performance of WGAN-fuzzing under various scenarios. Figure 7 demonstrates the FORR results of the general fuzzer, GAN-Fuzzer, and WGAN-Fuzzer. The results show that the WGAN-Fuzzing model generates a high percentage of realistic output samples compared to the other two algorithms. The WGAN model in NRS can improve output generation accuracy by fully automating the vulnerability analysis in I4.0 systems. For example, the general fuzzer, GAN-Fuzzer, and WGAN-Fuzzer accomplish 91%, 94.5% and 99.1% of FORR for 100 epochs. Figure 8 shows the TEV comparison results of general fuzzer, GAN-Fuzzer, and WGAN-Fuzzer. The WGAN-Fuzzing model increases the TEV by varying the epochs from low to high. The WGAN model can increase vulnerability analysis to its nature. Thus, it triggers the number of vulnerabilities in fuzzy output generation and increases the performance of the fuzzer. Moreover, Figure 9 demonstrates the DGD values of the general fuzzer, GAN-Fuzzer, and WGAN-Fuzzer by varying the epochs from 20 to 100. Compared to other fuzzers, the WGAN-fuzzer can improve the data diversity maintenance efficiency. For instance, general fuzzer, GAN-Fuzzer, and WGAN-Fuzzer attain 4.1, 6.3, and 7.8 of DGD under 100 epochs scenario.

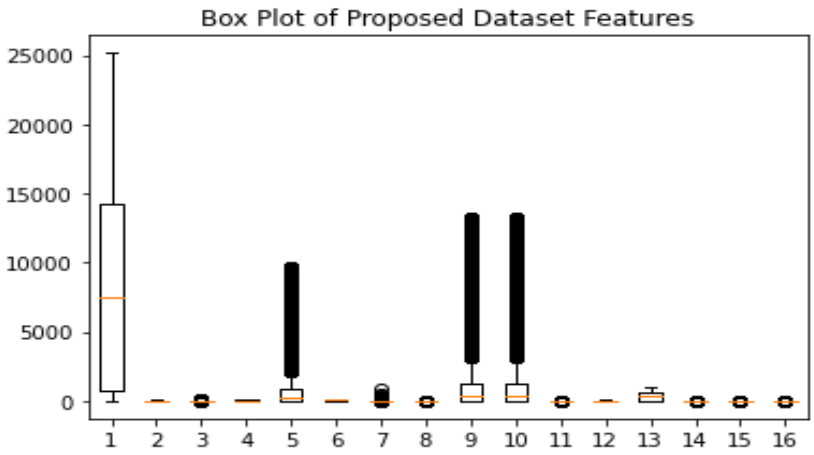


Figure 10: Box Plot for Proposed Dataset

Figure 10 plots the box plot of the proposed dataset. The box plot is a data visualization strategy that demonstrates the sixteen features' data distribution by employing multiple lines and boxes. It allows WGAN-fuzzer output dataset features over a geographical area. It displays the variation in samples of a numeric

fuzzer output dataset. The results show that the first feature is highly distributed, and features 9 and 10 receive second place for data distribution. The other features have minimum samples compared to 1, 9, and 10.

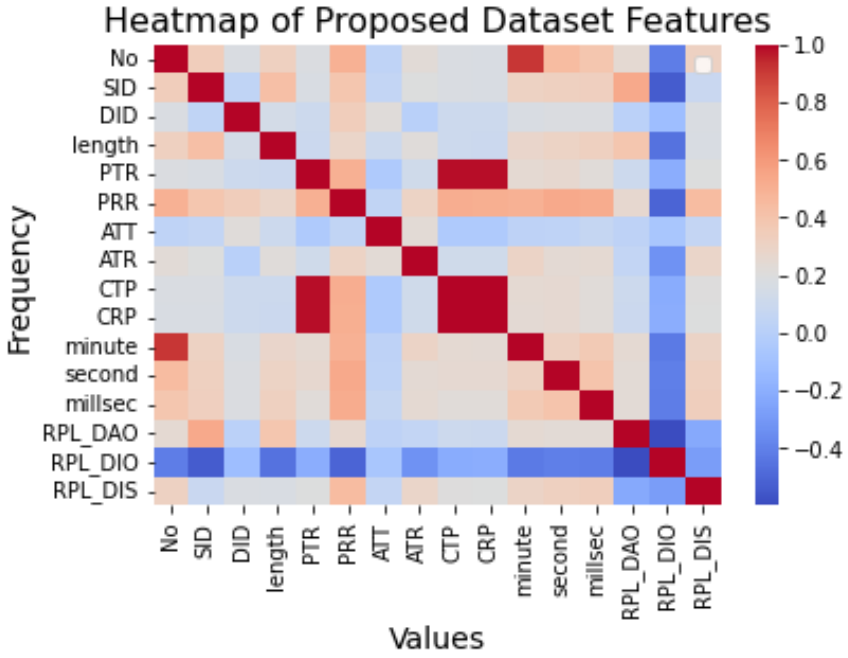
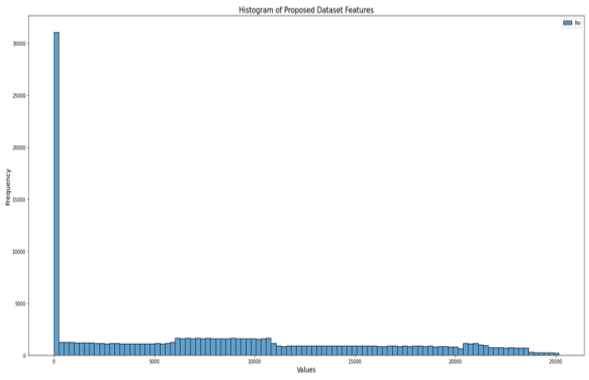
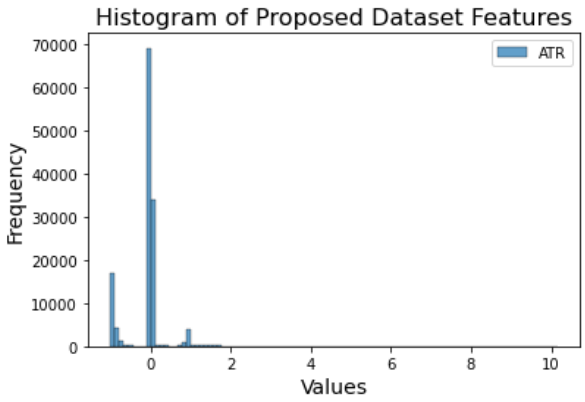


Figure 11: Heatmap of Proposed Dataset

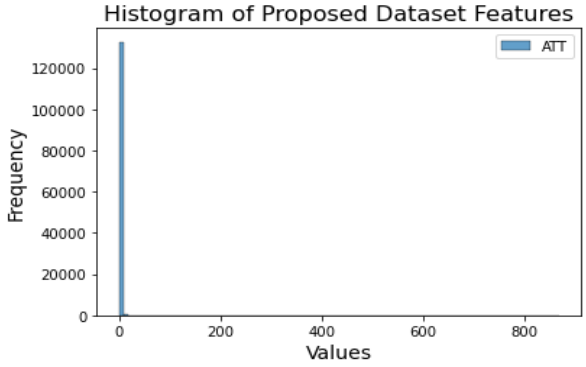
Figure 11 depicts the heatmap of the proposed dataset. The heatmap is a data visualization strategy representing the individual feature magnitudes within the WGAN-Fuzzer output dataset as color. The variations may be in two cases: hue and intensity. The heatmap assists in building the learning models with rich features that highly correlate to vulnerability detection. It is built by converting the correlation matrix into different color dimensions. The figures depict that CTP and CRP's feature space is highly rich for vulnerability detection. The dimensional space with one intensity value highly contributes to vulnerability detection.



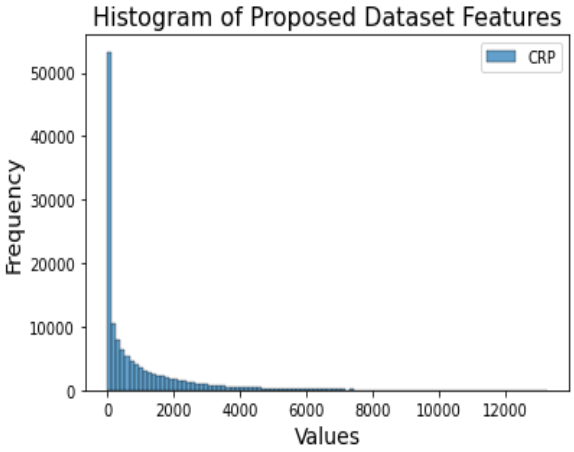
A



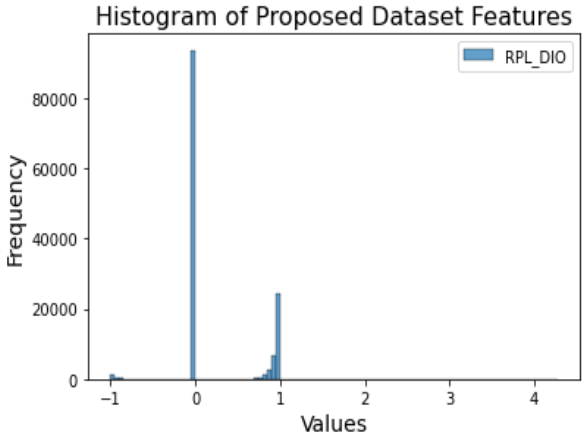
B



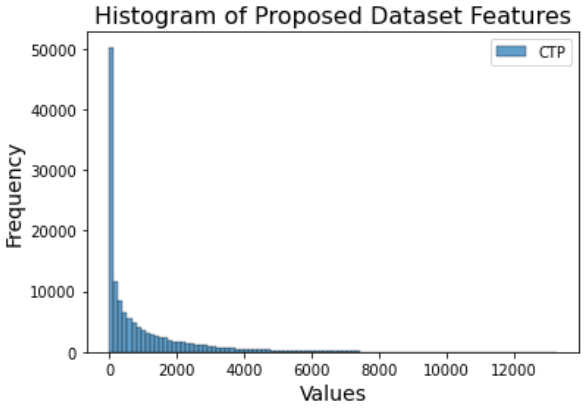
C


Histogram of Proposed Dataset Features
CRP
Frequency
Values
0
10000
20000
30000
40000
50000
0
2000
4000
6000
8000
10000
12000

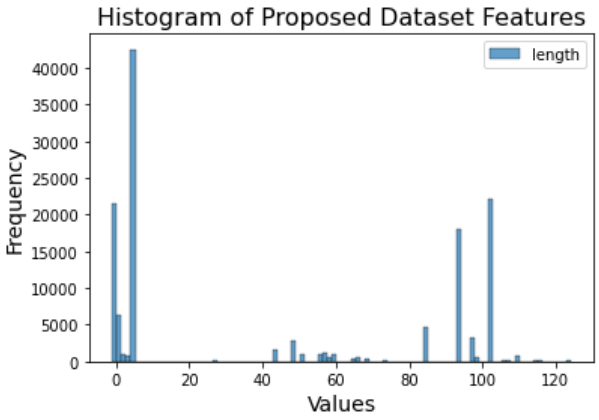
D

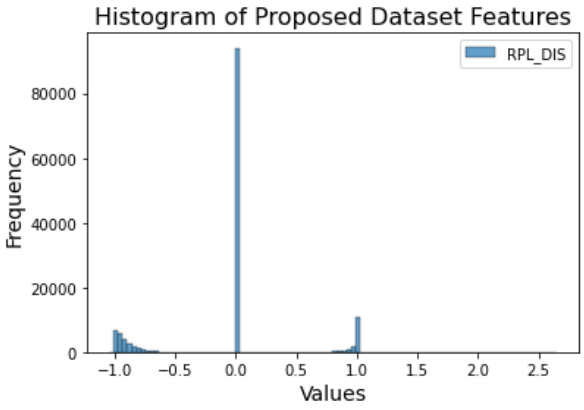

Histogram of Proposed Dataset Features
RPL_DIO
Frequency
Values
0
20000
40000
60000
80000
−1
0
1
2
3
4

G

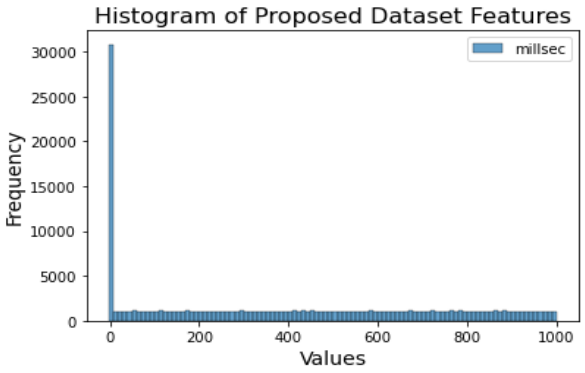

Histogram of Proposed Dataset Features
CTP
Frequency
Values
0
10000
20000
30000
40000
50000
0
2000
4000
6000
8000
10000
12000

E


Histogram of Proposed Dataset Features
length
Frequency
Values
0
5000
10000
15000
20000
25000
30000
35000
40000
0
20
40
60
80
100
120

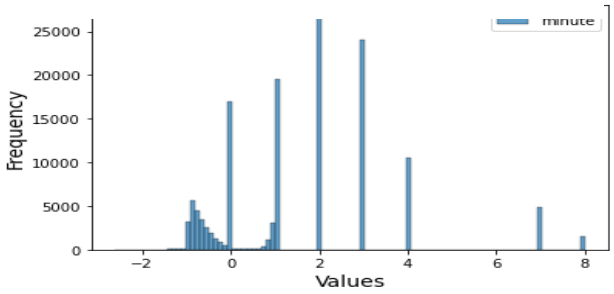

Histogram of Proposed Dataset Features
RPL_DIS
Frequency
Values
0
20000
40000
60000
80000
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
2.5

F

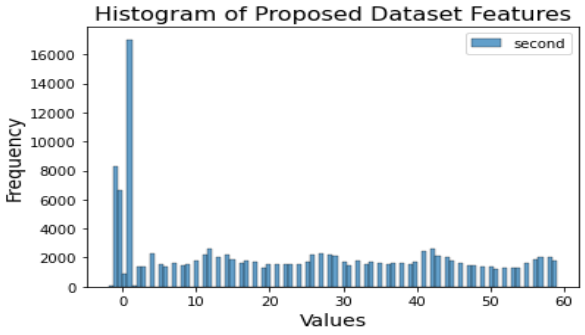


I

H



J



K

Figure 12: (A, B, C, D, E, F, G, H, I, J, and K) Histograms of Various Features of Proposed Dataset

Figure 12 compares the histogram frequencies of different features in the proposed dataset. The histogram representation assists in easily understanding which feature is most and least common for vulnerability detection. In a histogram representation, the feature type is shown on the horizontal axis, and the feature frequency is depicted on the vertical axis. The histograms of different features, including RPL-DIS, DAO, and DIO, show the importance of each feature in vulnerability analysis and attack detection.

Attack Discovery Results:

The NRS analyzes the attack discovery results of the proposed VTAD mechanism by comparing it with existing deep learning models that are Random Forest (RF) [38], Recurrent Neural Networks (RNN) [39], and Convolutional Neural Networks (CNN) [40], CNN-Image (CNN-I), and Gated Recurrent Unit (GRU). For evaluation, the fuzzer output dataset that comprises numeric data is provided as input to the RF, RNN, and CNN algorithms. Instead, the converted images are provided as input to the CNN-I, GRU, and proposed VTAD. Moreover, the performance of RF, RNN, and CNN are evaluated with numeric datasets, and the efficacy of CNN-I, GRU, and VTAD are analyzed using image datasets.

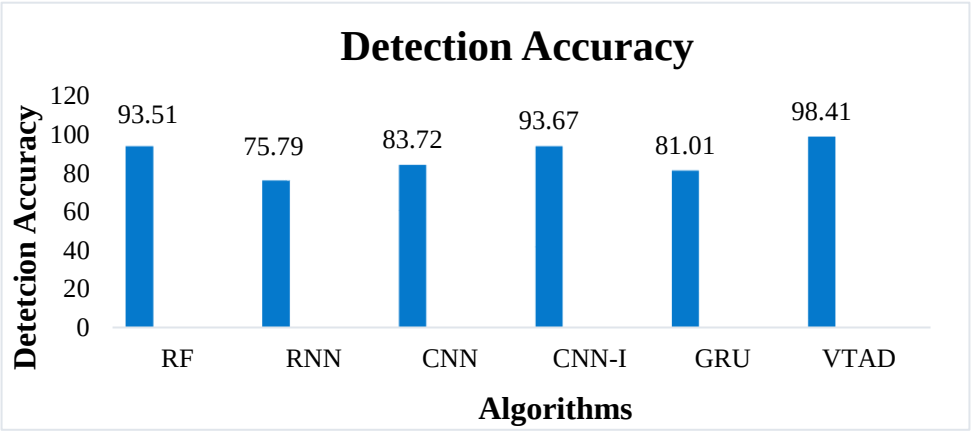


Figure 13: Algorithms Vs Detection Accuracy

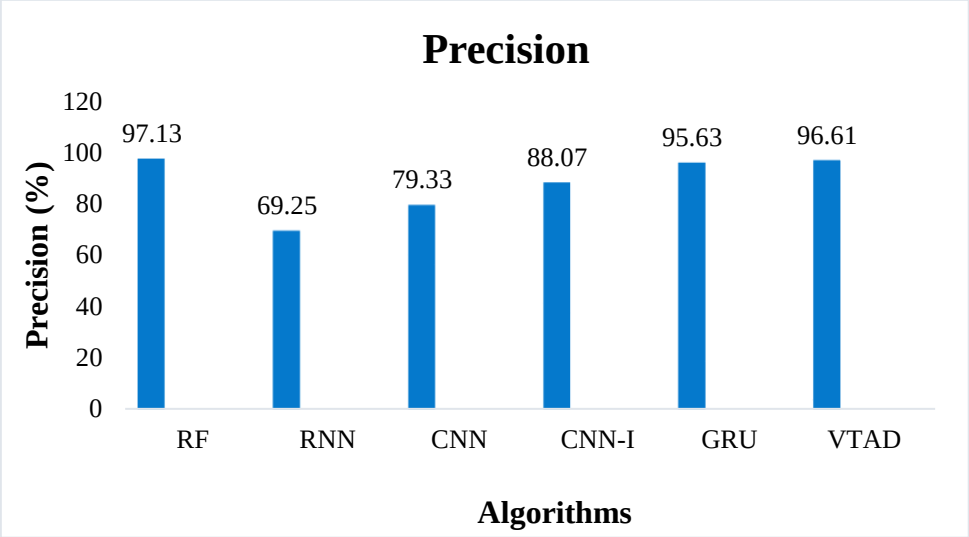


Figure 14: Algorithms Vs Precision

Figures 13 and 14 show the detection accuracy and precision results of RF, RNN, CNN, CNN-I, GRU, and proposed VTAD algorithms. In Figure 10, the proposed VTAD outperforms the other five algorithms in terms of accuracy. For instance, the detection accuracy of VTAD is 98.41%. It is higher by 4.9%, 22.62%, 14.69%, 4.74%, and 17.4% than RF, RNN, CNN, CNN-I, and GRU, respectively. The main reason is that the proposed VTAD exploits the advantage of ViT in vulnerability detection in which the complex patterns are effectively learned from a high-dimensional I4.0 dataset. Thus, the proposed vulnerability detection strategy VTAD minimizes the error rates and increases the detection accuracy compared to the other five algorithms. For example, the precision of VTAD is 96.61%, whereas the GRU accomplishes 95.63% of precision. By converting the numeric dataset into the image, the VTAD improves the learning accuracy and enhances the precision rate.

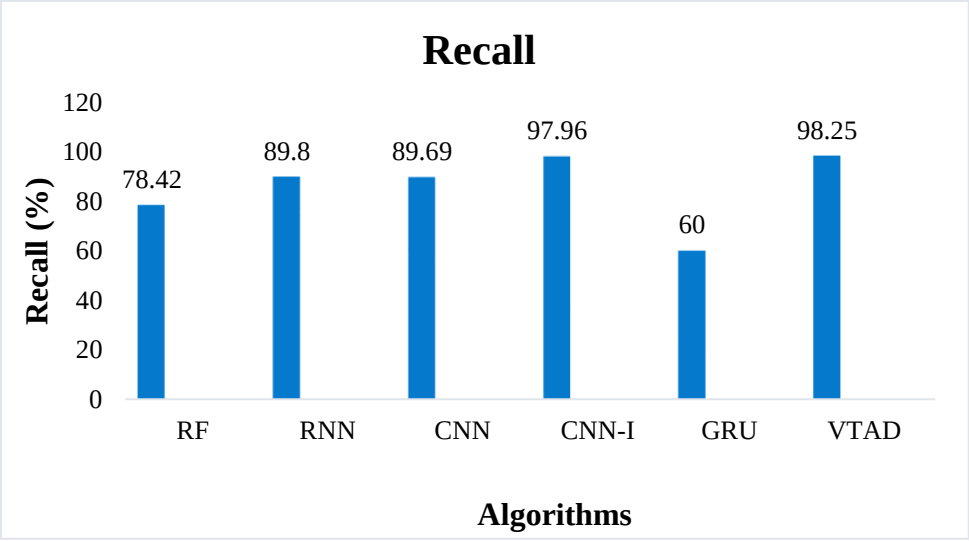


Figure 15: Algorithms Vs. Recall

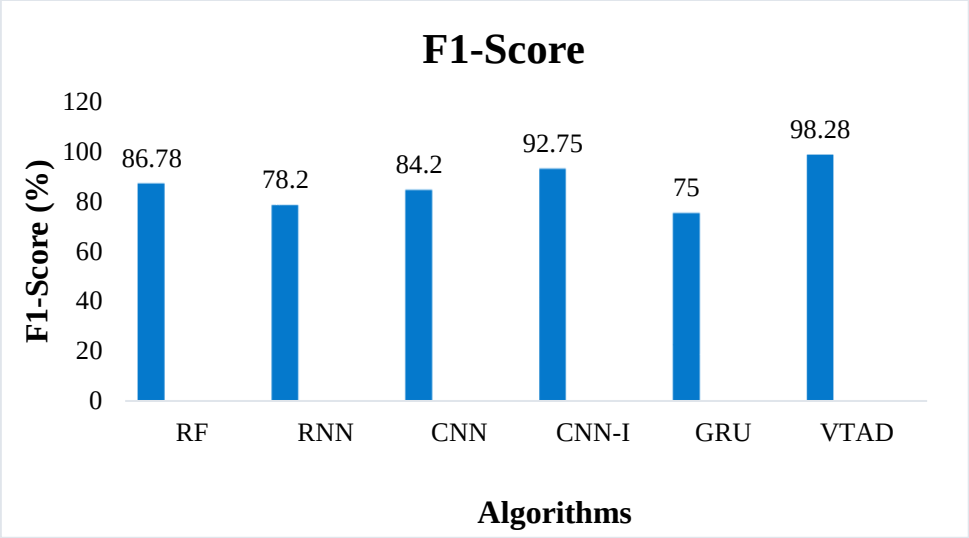
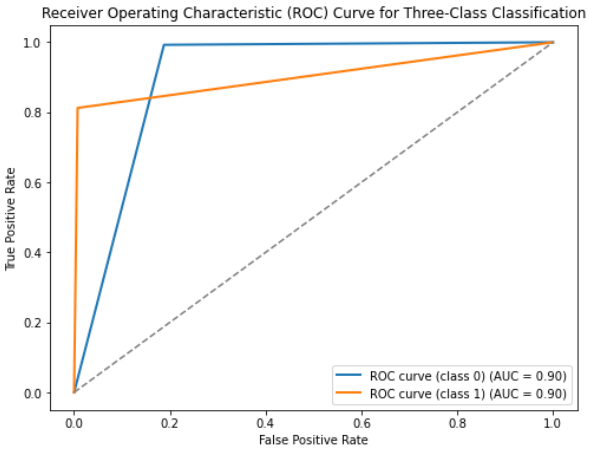
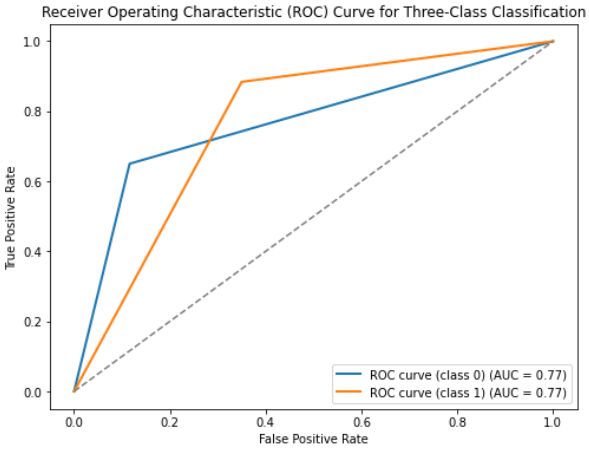


Figure 16: Algorithms Vs F1-Score

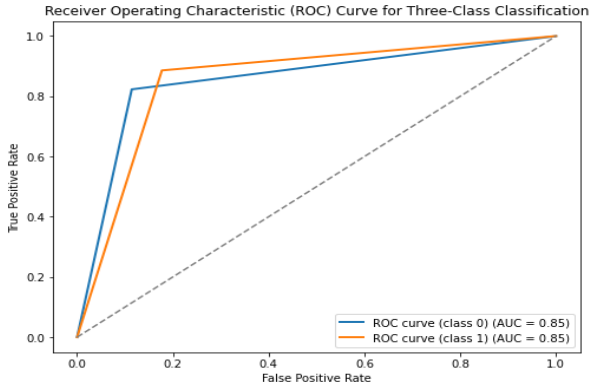
Figures 15 and 16 depict the comparative relationship of recall and F1-score of RF, RNN, CNN, CNN-I, GRU, and proposed VTAD algorithms. The results demonstrate that the proposed VTAD model accomplishes better than the other five algorithms. The main reason is that the VTAD model exploits the ViT that precisely captures the spatial relationship among the features in the image dataset and improves the learning efficiency, resulting in a high vulnerability detection rate. Thus, it boosts the recall when compared with other algorithms. For example, the proposed model obtains 98.25% of recall value, improving by 19.83%, 8.45%, 8.56%, 0.29% and 38.25% to RF, RNN, CNN, CNN-I, and GRU algorithms. Unlike VTAD, the RF, RNN, and CNN do not incorporate the ViT and fail to effectively capture the complex patterns and subtle deviation in the large-scale I4.0 dataset. Thus, it minimizes the F1-score when compared with VTAD. For instance, the RF, CNN, and VTAD attain 86.78%, 84.2%, and 98.28% of F1-score values, respectively.



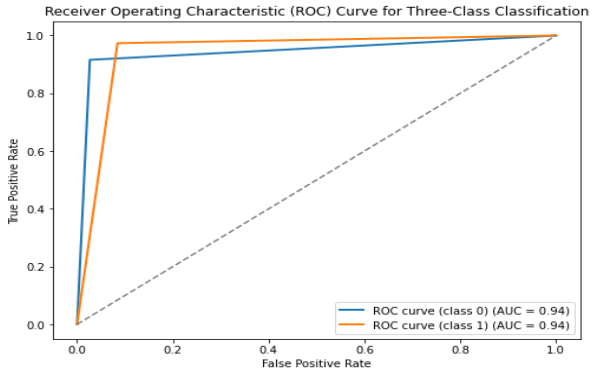
RF



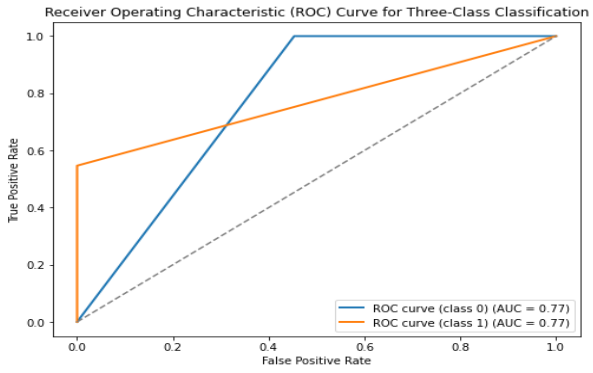
RNN



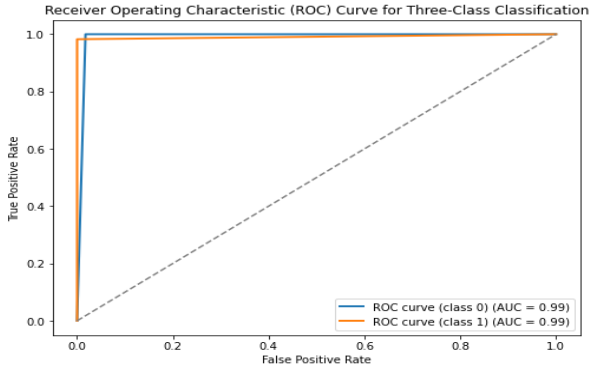
CNN



CNN-I



GRU



VTAD

Figure 17: Illustrates the ROC curve relationships of different learning strategies.

RF, RNN, CNN, CNN-I, GRU, and VTAD. Generally, the ROC curves show the visualization trade-off between the false positive rate (x-axis) and the true positive rate (y-axis) across different decision thresholds. The proposed NRS estimates the false positive and true positive rates for each learning strategy RF, RNN, CNN, CNN-I, GRU, and VTAD across various thresholds for constructing the ROC curve visualization. Through visual inspections, it is concluded that the ROC curves of the proposed VTAD are closer to the plot's top-left corner, which is more desirable, as it accomplishes a higher true positive rate value while maintaining a lower false positive rate value, as shown in figure 17 (f).

Conclusions

This paper proposes a Novel RPL Security NRS model for the RPL-based I4.0 manufacturing environment. Two techniques, such as WFVA and VTAD, are integrated by NRS for vulnerability analysis and discovery. The NRS collects the realistic RPL-IIoT data and constructs the novel RID to rectify the dataset lacking issues. The proposed NRS effectively captures the complex patterns and improves the learning efficiency by converting the numeric dataset into an image dataset. Further, it escalates the system automata and minimizes the resource consumption costs of the I4.0 environment by integrating a WGAN-Fuzzing-based vulnerability analysis model. The algorithm complexity is reduced by adjusting the hidden layers according to the dataset size. The NRS enhances the detection accuracy level by providing the high vulnerability distributed image-based fuzzer output data as input to the ViT-based attack discovery model, resulting in high vulnerability detection efficiency. Moreover, the simulation results demonstrate that the proposed NRS outperforms the existing methods regarding fuzzer output recognition rate, triggered efficiency of vulnerabilities, diversity of generated data, detection accuracy, precision, recall, and F1-score.

References

[1] Malik, P. K., Sharma, R., Singh, R., Gehlot, A., Satapathy, S. C., Alnumay, W. S., ... & Nayak, J. (2021). Industrial Internet of Things and its applications in industry 4.0: State of the art. Computer Communications, 166, 125-139.

[2] Alcácer, V., & Cruz-Machado, V. (2019). Scanning the industry 4.0: A literature review on technologies for manufacturing systems.

Engineering science and technology, an international journal, 22(3), 899-919.

[3] Darabkh, K. A., Al-Akhras, M., Zomot, J. N., & Atiquzzaman, M. (2022). RPL routing protocol over IoT: A comprehensive survey, recent advances, insights, bibliometric analysis, recommendations, and future directions. Journal of Network and Computer Applications, 207, 103476.

[4] Berger, S., Bürger, O., & Röglinger, M. (2020). Attacks on the Industrial Internet of Things–Development of a multi-layer Taxonomy. Computers & Security, 93, 101790.

[5] Altulaihan, E., Almaiah, M. A., & Aljughaiman, A. (2022). Cybersecurity threats, countermeasures and mitigation techniques on the IoT: future research directions. Electronics, 11(20), 3330.

[6] Raimundo, R. J., & Rosário, A. T. (2022). Cybersecurity in the internet of things in industrial management. Applied Sciences, 12(3), 1598.

[7] Aslan, Ö., Aktuğ, S. S., Ozkan-Okay, M., Yilmaz, A. A., & Akin, E. (2023). A comprehensive review of cyber security vulnerabilities, threats, attacks, and solutions. Electronics, 12(6), 1333.

[8] Northern, B., Burks, T., Hatcher, M., Rogers, M., & Ulybyshev, D. (2021). Vercasm-cps: Vulnerability analysis and cyber risk assessment for cyber-physical systems. Information, 12(10), 408.

[9] Eceiza, M., Flores, J. L., & Iturbe, M. (2021). Fuzzing the internet of things: A review on the techniques and challenges for efficient vulnerability discovery in embedded systems. IEEE Internet of Things Journal, 8(13), 10390-10411.

[10] Böhme, M., Cadar, C., & Roychoudhury, A. (2020). Fuzzing: Challenges and reflections. IEEE Software, 38(3), 79-86.

[11] Jeon, S., & Kim, H. K. (2021). AutoVAS: An automated vulnerability analysis system with a deep learning approach. Computers & Security, 106, 102308.

[12] Yang, S., Chen, W., Zhang, X., & Yang, W. (2021). A graph-based method for vulnerability analysis of renewable energy integrated power systems to cascading failures. Reliability Engineering & System Safety, 207, 107354.

[13] Li, S., Ding, T., Jia, W., Huang, C., Catalão, J. P., & Li, F. (2021). A machine learning-based vulnerability analysis for cascading failures of integrated power-gas systems. IEEE Transactions on power systems, 37(3), 2259-2270.

[14] Li, X., Wang, L., Xin, Y., Yang, Y., Tang, Q., & Chen, Y. (2021). Automated software vulnerability detection based on hybrid neural network. Applied Sciences, 11(7), 3201.

[15] Williams, M. A., Barranco, R. C., Naim, S. M., Dey, S., Hossain, M. S., & Akbar, M. (2020). A vulnerability analysis and prediction framework. Computers & Security, 92, 101751.

[16] Wu, Y., Zou, D., Dou, S., Yang, W., Xu, D., & Jin, H. (2022, May). VulCNN: An image-inspired scalable vulnerability detection system. In Proceedings of the 44th International Conference on Software Engineering (pp. 2365-2376).

[17] Hin, D., Kan, A., Chen, H., & Babar, M. A. (2022, May). LineVD: Statement-level vulnerability detection using graph neural networks. In Proceedings of the 19th International Conference on Mining Software Repositories (pp. 596-607).

[18] Koroniotis, N., Moustafa, N., Turnbull, B., Schiliro, F., Gauravaram, P., & Janicke, H. (2021, October). A deep learning-based penetration testing framework for vulnerability identification in internet of things environments. In 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom) (pp. 887-894). IEEE.

[19] Foley, J., Moradpoor, N., & Ochen, H. (2020). Employing a machine learning approach to detect combined internet of things attacks against two objective functions using a novel dataset. Security and Communication Networks, 2020, 1-17.

[20] Al-Boghdady, A., El-Ramly, M., & Wassif, K. (2022). iDetect for vulnerability detection in internet of things operating systems using machine learning. Scientific Reports, 12(1), 17086.

[21] Rajawat, A. S., Rawat, R., Barhanpurkar, K., Shaw, R. N., & Ghosh, A. (2021). Vulnerability analysis at industrial internet of things platform on dark web network using computational intelligence. Computationally intelligent systems and their applications, 39-51.

[22] Huang, Y., Jiang, B., & Chan, W. K. (2020, November). EOSFuzzer: Fuzzing eosio smart contracts for vulnerability detection. In Proceedings of the 12th Asia-Pacific Symposium on Internetware (pp. 99-109).

[23] Wang, X., Sun, J., Hu, C., Yu, P., Zhang, B., & Hou, D. (2022). EtherFuzz: Mutation Fuzzing Smart Contracts for TOD Vulnerability Detection. Wireless Communications and Mobile Computing, 2022.

[24] Liu, Z., Fang, Y., Huang, C., & Xu, Y. (2022). GAXSS: effective payload generation method to detect XSS vulnerabilities based on genetic algorithm. Security and Communication Networks, 2022, 1-15.

[25] Blair, W., Mambretti, A., Arshad, S., Weissbacher, M., Robertson, W., Kirda, E., & Egele, M. (2020). HotFuzz: Discovering algorithmic denial-of-service vulnerabilities through guided micro-fuzzing. arXiv preprint arXiv:2002.03416.

[26] Zhu, X., Wen, S., Jolfaei, A., Haghighi, M. S., Camtepe, S., & Xiang, Y. (2020). Vulnerability detection in SIoT applications: A fuzzing method on their binaries. IEEE Transactions on Network Science and Engineering, 9(3), 970-979.

[27] Casteur, G., Aubaret, A., Blondeau, B., Clouet, V., Quemat, A., Pical, V., & Zitouni, R. (2020, June). Fuzzing attacks for vulnerability discovery within MQTT protocol. In 2020 International Wireless Communications and Mobile Computing (IWCMC) (pp. 420-425). IEEE.

[28] Zeng, Y., Lin, M., Guo, S., Shen, Y., Cui, T., Wu, T., ... & Wang, Q. (2020). Multifuzz: A coverage-based multiparty-protocol fuzzer for iot publish/subscribe protocols. Sensors, 20(18), 5194.

[29] Kwon, S., Son, S. J., Choi, Y., & Lee, J. H. (2021). Protocol fuzzing to find security vulnerabilities of RabbitMQ. Concurrency and Computation: Practice and Experience, 33(23), e6012.

[30] Kim, S., Cho, J., Lee, C., & Shon, T. (2020). Smart seed selection-based effective black box fuzzing for IIoT protocol. The Journal of Supercomputing, 76, 10140-10154.

[31] Kolisnyk, M. (2021). Vulnerability analysis and method of selection of communication protocols for information transfer in Internet of Things systems. Radioelectronic and computer systems, (1), 133-149.

[32] Park, J. I., & Hong, C. H. (2023). VFP: Converting Tabular Data for IIoT into Images Considering Correlations of Attributes for Convolutional Neural Networks. arXiv preprint arXiv:2303.09068.

[33] Aslan, M. F., Sabanci, K., & Durdu, A. (2021). A CNN-based novel solution for determining the survival status of heart failure patients with clinical record data: numeric to image. Biomedical Signal Processing and Control, 68, 102716.

[34] Uparkar, O., Bharti, J., Pateriya, R. K., Gupta, R. K., & Sharma, A. (2023). Vision Transformer Outperforms Deep Convolutional Neural Network-based Model in Classifying X-ray Images. Procedia Computer Science, 218, 2338-2349.

[35] Bazi, Y., Bashmal, L., Rahhal, M. M. A., Dayil, R. A., & Ajlan, N. A. (2021). Vision transformers for remote sensing image classification. Remote Sensing, 13(3), 516.

[36] Du, X., Chen, A., He, B., Chen, H., Zhang, F., & Chen, Y. (2022). AflIot: Fuzzing on linux-based IoT device with binary-level instrumentation. Computers & Security, 122, 102889.

[37] Ye, A., Wang, L., Zhao, L., Ke, J., Wang, W., & Liu, Q. (2021). Rapidfuzz: Accelerating fuzzing via generative adversarial networks. Neurocomputing, 460, 195-204.

[38] Choubisa, M., Doshi, R., Khatri, N., & Hiran, K. K. (2022, May). A simple and robust approach of random forest for intrusion detection system in cyber security. In *2022 International Conference on IoT and Blockchain Technology (ICIBT)* (pp. 1-5). IEEE.

[39] Almiani, M., AbuGhazleh, A., Al-Rahayfeh, A., Atiewi, S., & Razaque, A. (2020). Deep recurrent neural network for IoT intrusion detection system. Simulation Modelling Practice and Theory, 101, 102031.

[40] Lawrence, T., & Zhang, L. (2019). IoTNet: An efficient and accurate convolutional neural network for IoT devices. Sensors, 19(24), 5541.