Computers, Materials & Continua
DOI:10.32604/cmc.2022.020826
Article

Tech Science Press

Efficient Autonomous Defense System Using Machine Learning on Edge Device

Jaehyuk Cho*

School of Electronic Engineering, Soongsil University, Seoul, 06978, Korea
*Corresponding Author: Jaehyuk Cho. Email: chojh@ssu.ac.kr, jaehyukcho7@gmail.com
Received: 10 June 2021; Accepted: 11 July 2021

Abstract: As a large amount of data needs to be processed and speed needs to be improved, edge computing with ultra-low latency and ultra-connectivity is emerging as a new paradigm. These changes can lead to new cyber risks, and should therefore be considered for a security threat model. To this end, we constructed an edge system to study security in two directions, hardware and software. First, on the hardware side, we want to autonomically defend against hardware attacks such as side channel attacks by configuring field programmable gate array (FPGA) which is suitable for edge computing and identifying communication status to control the communication method according to priority. In addition, on the software side, data collected on the server performs end-to-end encryption via symmetric encryption keys. Also, we modeled autonomous defense systems on the server by using machine learning which targets to incoming and outgoing logs. Server log utilizes existing intrusion detection datasets that should be used in real-world environments. Server log was used to detect intrusion early by modeling an intrusion prevention system to identify behaviors that violate security policy, and to utilize the existing intrusion detection data set that should be used in a real environment. Through this, we designed an efficient autonomous defense system that can provide a stable system by detecting abnormal signals from the device and converting them to an effective method to control edge computing, and to detect and control abnormal intrusions on the server side.

Keywords: Autonomous defense; side channel attack; intrusion prevention system; edge computing; machine learning

1 Introduction

Centralized computing structures have large data volumes such as smart factories, smart farms, and self-driving cars. Also, there is a problem to ask for real-time processing, such as overloading cloud servers due to network traffic. To solve this problem, edge computing technologies that process data in real-time or near-field are gaining attention [1]. However, as the utilization of edge computing facilitates smart convergence across the infrastructure, the surface of cybersecurity expands [2–4]. It also increases the threat posed by cyberattacks such as system compromise, service interruption, and information leakage [5,6]. In particular, the possibility of cyberattacks on drones and self-driving cars along with IoT is expanding beyond the scope of

existing cybersecurity areas such as information corruption, leakage, and service interference [7,8]. It is even expanding into areas where safety of human life and property must be guaranteed. Security technologies for these cyberattacks have evolved primarily on a software basis and hacking incidents at various institutions have raised significant problems with software security systems [9,10]. Recently, various hardware security systems have been proposed to complement the limitations of these software security systems. In this paper, we conduct a two-direction study on the security of edge computing. Security at the field programmable gate array (FPGA) in hardware and Intrusion Prevention System (IPS) on software basis was introduced. Through this, we propose ways to enhance the security of the system.

First, in hardware, security in FPGA was studied. With increasing reliance on hardware acceleration to improve the performance and energy efficiency of computing systems, FPGA has recently been widely adopted as a system that has to handle large amounts of data, such as edge computing. Since such FPGA is designed by hardware manufacturers, third parties, consignees, etc., there is a possibility of being attacked for malicious purposes during the development and deployment phase of the FPGA [11]. Especially, in edge computing, many edge devices are located outside the secure data center. So physical security levels are not the same as servers in the data center. If an attack is made against such an FPGA, the actual system could suffer damage such as a service outage. In addition, when FPGA is adopted in the cloud and data center or integrated into system-on-a-chip (SoC), an attacker may perform remote side channel attacks without physical access or close proximity to hardware. To complement this, we study to increase the security of hardware by detecting power anomalies, signal anomalies, and clock(clk) anomalies in FPGA and controlling communication methods according to priorities.

Second, for software, we study IPS methods and apply them to security. Edge computing devices with insufficient configuration or security provide an opportunity for attackers to disrupt operations and gain extensive access to enterprise networks [12]. Computing and storage capabilities mounted on edge servers are being strengthened, connected to network, and handling sensitive data for many reasons. An attacker can target this system to steal data or use it as a springboard for other attacks by such as Distributed Denial of Service (DDoS) attack. In particular, basic security mistakes, such as using pre-set passwords or deploying systems without multiple authentication, can have significant consequences.

Therefore zero-trust and anomaly detection capabilities are critical in edge computing environments [13]. To this end, we study the IPS scheme to prevent unauthorized users from accessing the system and to protect the information. We detect suspicious network activity while monitoring already known attack signatures as well as restrictive blocking of known attacks. It finds an attack signature that might have found suspicious network activity during the detection process. We then observe the traffic on the network and extract the feature for the attack signature from IPS's method on packets with suspicious activity. Finally, it detects the attack method and is designed to be autonomous defense by using AI.

2 Materials and Methods

As shown in Fig. 1, developed IoT process network system consists of devices, network and cloud. Where device part is designed with edge controller and network includes WiFi, Long Range (LoRa), Narrowband-Internet of Things (NB-IoT), Long Term Evolution (LTE). Cloud integrates data collection, process and analysis. Intrusion prevention system is applied to deal with security problems in OSI (Open Systems Interconnection) network layer [14]. This session introduces the device part, the network part, and finally the experimental setup.

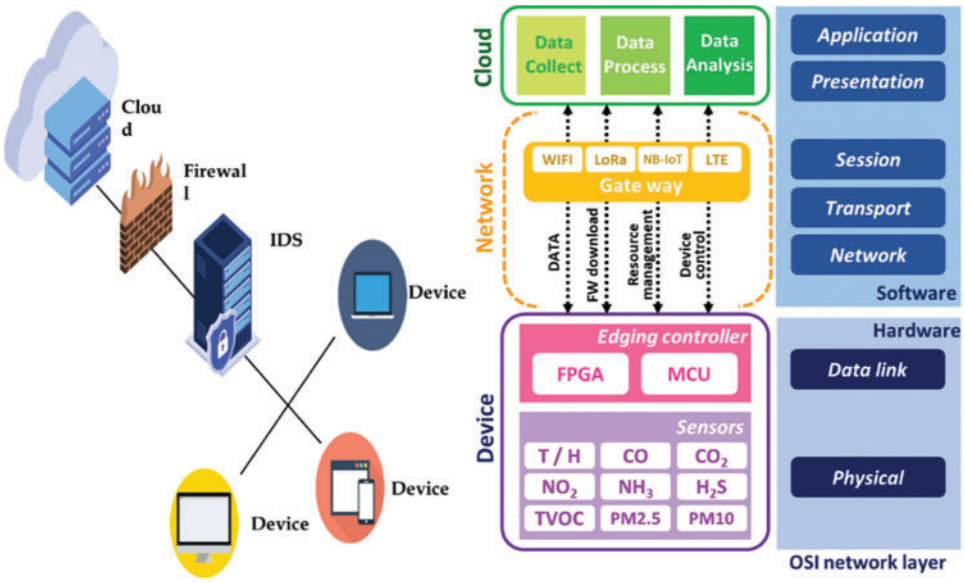

Figure 1: Architecture of IoT system comparing OSI network layer

2.1 Security in Hardware

2.1.1 FPGA with Edge Device

FPGA is a device that combines some of the advantages of software such as rapid development, low initial technology cost, no repetitive additional costs, hardware advantages of high performance and high-power efficiency. In particular, FPGA can adapt to all algorithmic properties due to hardware flexibility, which is different from Central Processing Unit (CPU) and Graphics Processing Unit (GPU) that mainly utilize spatial parallel processing. Thus, FPGA is adopted as a model suitable for edge computing, with more granular and large-scale utilization of spatial and temporal parallel processing. In addition, GPU and CPU are optimized for batch processing of memory in applications that require large amounts of data processing, such as embedded system design, routing, signal processing, or encryption. On the other hand, FPGA is inherently appropriate to accelerate streaming applications. A pipeline streaming architecture with data flow control is easily built on the FPGA to handle data and command streams and generate output results at a constant throughput. Above all, the most important characteristics that distinguish FPGA from other semiconductors are cost, flexibility, and heterogeneous parallel processing. Compared to Application Specific Integrated Circuit (ASIC) and GPU, there is no additional power consumption cost due to low heat generation problems. And it can be responded only by changing logic design with greater flexibility. In addition, unlike CPUs, it has heterogeneous parallelism and plays a key role in artificial intelligence technology with fast computation [15].

The FPGA chip has Optional Mask Programmable Memory (MPM) and supports LVDS 600 Mbps per lane with up to 6 TX pairs and 6 RX pairs. And this board can synthesize clock signals using PLL. The operation method of FPGA and Microcontroller Unit (MCU) is explained in terms of Local Area Network (LAN) selection algorithm, anomaly correction, firmware management, resource management and device management [16].

2.1.2 Types of Attacks in FPGAs

Hardware Trojan (HT) in FPGA can be introduced through multiple hardware designers, third-party intelligent property (3PIP) providers, and outsourcing companies during the

development phase. HT typically consists of triggers and payloads [17]. Triggers activate payloads when certain conditions are met. Then payloads are inserted into the circuit in the malicious behavior intended by the attacker to perform malicious functions such as Denial of Service, functional changes, degradation, and sensitive information leakage. Unlike payloads distributed across multiple modules, triggers are implemented in one module and complex conditions are used to prevent payloads from being activated during the test phase [18]. These HTs allow attackers to insert HT into the entire development process of FPGA, i.e., design, synthesis, and deployment. Also, RTL and 3PIP can be converted to a batch and 3PIP completed netlist, and outsourcing companies provide hardware design in netlist form.

Reverse engineering mainly includes gate-level netlist reverse engineering and image processing-based reverse engineering by removing packages of chips and each layer of chips, taking layouts and analyzing computer-acquired layouts [19]. In gate-level netlist reverse engineering, an attacker can extract high-level feature information from a gate-level netlist, such as Register Transfer Level (RTL) or structural-level descriptions. X-ray tomography is a non-destructive method that can provide a layer-specific image of a chip and is often used in the analysis of internal via, traces, wire bonding, capacitors, contacts, or resistors. The destructive method is to etch and grind all layers. It can obtain all layout information inside the chip, and depending on the technology, it is possible to remove only external packages while the chip is operating and observe the signal wherever you want. Intellectual Property (IPs) such as memory and cores are easily distinguishable when internal layouts are acquired by reverse engineering. So, buses connected to these IPs can also be easily found. Reverse engineering makes it possible to observe the bus inside the SoC and extract data traveling through the bus [20]. In a similar way, not only internal code but also moving key values can be extracted via bus. Especially, there are vulnerabilities that can easily decode internal secret information or falsify signatures when a key is exposed.

Side channel attack is a method of attacking information such as power, signal, heat, and computational time that changes when a chip is operating [21]. This attack can collect and analyze side channel information to extract key security information, such as secret data and keys. And it requires relatively little time and cost because it uses only interfaces that are accessible from the outside of the chip. Because of these features, side channel attacks are powerful non-invasive attacks that exploit the leakage of physical information when cryptographic algorithms run in a system. And side channel attacks are powerful attacks that leave no trace or destroy encryption devices because they exploit accessible data, clocks, and voltage interfaces on the target device [22,23]. Thus, side channel attacks have become a major threat to current cryptographic devices Fault injection attacks are one of the most effective sub-channel attacks, which use voltage glitches, clock glitches, and laser pulses to inject faults that interfere with the operational state of a cryptographic device or chip and generate controllable faults [24]. Incorrect information can be exploited to compute keys via differential encryption analysis.

2.1.3 LAN Network Selection Algorithm

Each consists of LAN network selection algorithms for high-level network access, and all LAN ports are connected to the Ethernet Physical layer (PHY) chip through FPGA. LAN network selection algorithms are composed of LoRa, WiFi, LTE, NB-IoT (5G capable) external modem.

The FPGA detects the activity of the 4 LAN ports in real time (alert detection). The FPGA available of high-speed operation is capable of high-speed data processing for several. By using

this feature, it is possible to detect failure of the operating LAN port at high speed. Detections include link detection, communication signal detection and power status detection.

As a countermeasure in case of a communication line problem, first, when a communication line problem is detected, automatic port transfer according to priority. When the line is changed, the current main line server and the failure detection line server are notified. During normal operation of 4 lines, it can be operated as a server at the same time, and when a failure of one line is detected, a MAC IC communication error is determined as a MAC (Media Access Controller) Integrated Circuit (IC) error, and a PHY IC communication error is determined as a PHY IC error.

Algorithm switching is divided into self-determination algorithm switching method and server algorithm switching. In the self-determination algorithm switching, when operating individual ports, the line priority is operated in the order of ports 1, 2, 3, and 4. Server Algorithm Switching is based on the artificial intelligence optimization algorithm of the server, and if it is necessary to change the current operation port, it is possible to change the operation port after receiving a server message. Fig. 2 shows the LAN network selection algorithm.

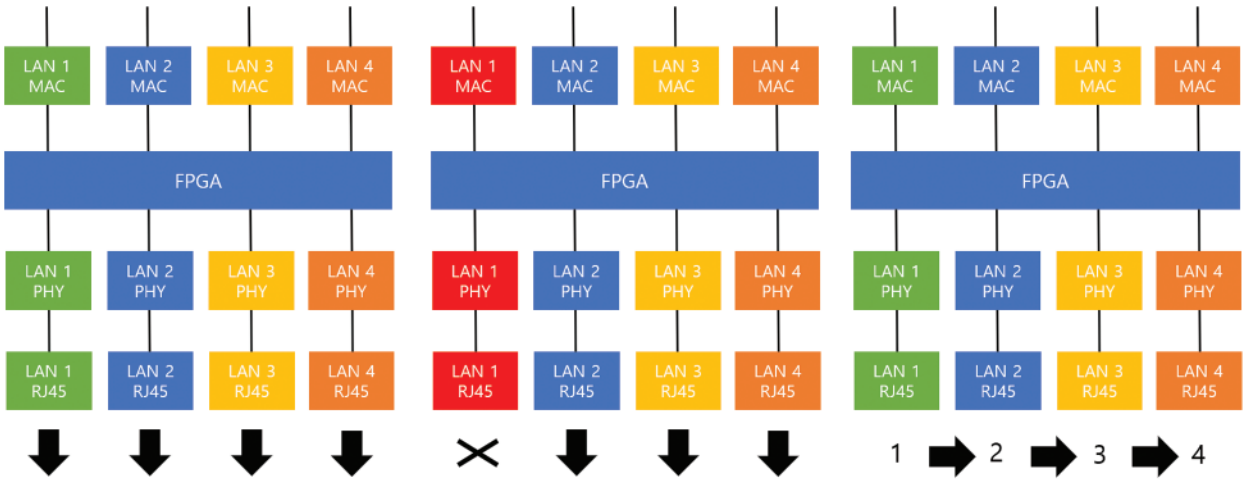

Figure 2: Algorithm of LAN network selection

2.1.4 Wifi, LTE, NB-IoT, Lora External Network Configuration

The external network connected to the device consists of four external networks: an external LTE modem, an external WiFi client, an NB-IoT external modem, and an external LoRa modem. To implement such a communication method, a signal connection is made through an FPGA and a power fault detection is monitored in real time using MCU, Analog-digital converter (ADC). NB-IoT is an Attention (AT) command method using RS-232, which does not require Media Access Control (MAC) or Physical layer (PHY) devices. By default, active networks are used, but when multiple networks are present, network priorities are operated in the order of WiFi, LTE, NB-IoT, and LoRa. Prioritization was determined by considering the stability of the network, the size of the Payload, and the cost.

2.1.5 Fault Detection

- Link Fault Detection

PHY chip and link connection status of WiFi and LTE external modems are notified to FPGA. NB-IoT does not have MAC or PHY, so if there is no response when AT command communicating with an external NB-IoT modem in the MCU, it determines that it is a link fault (external modem fault) and notifies FPGA as shown in Fig. 3.

- Communication Signal Fault Detection

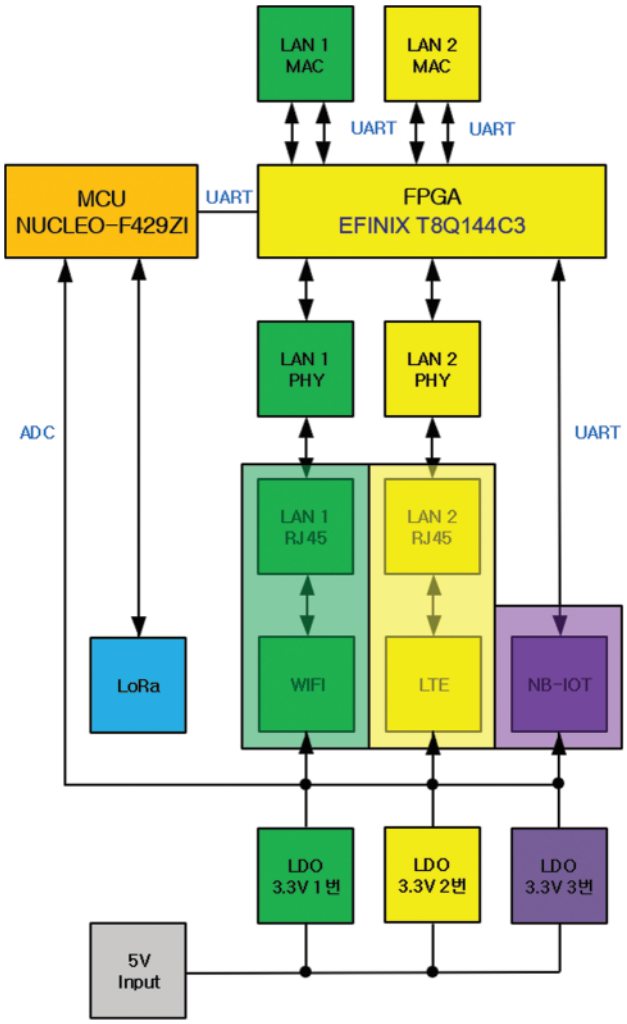

Figure 3: Link fault detection

All signals of LAN1 and LAN2 using MAC and PHY are connected by FPGA and the state of signals necessary for communication such as clock and control signals is detected in the FPGA in real time to determine whether there is a fault. The communication signal fault conditions are when there is no clock and the control signal is not operating in Fig. 4. LAN1 and LAN2 have a maximum speed signal of 50 MHz and detect real-time communication signal fault in 10nsec units using 100 MHz PLL inside the FPGA.

- Power Fault Detection

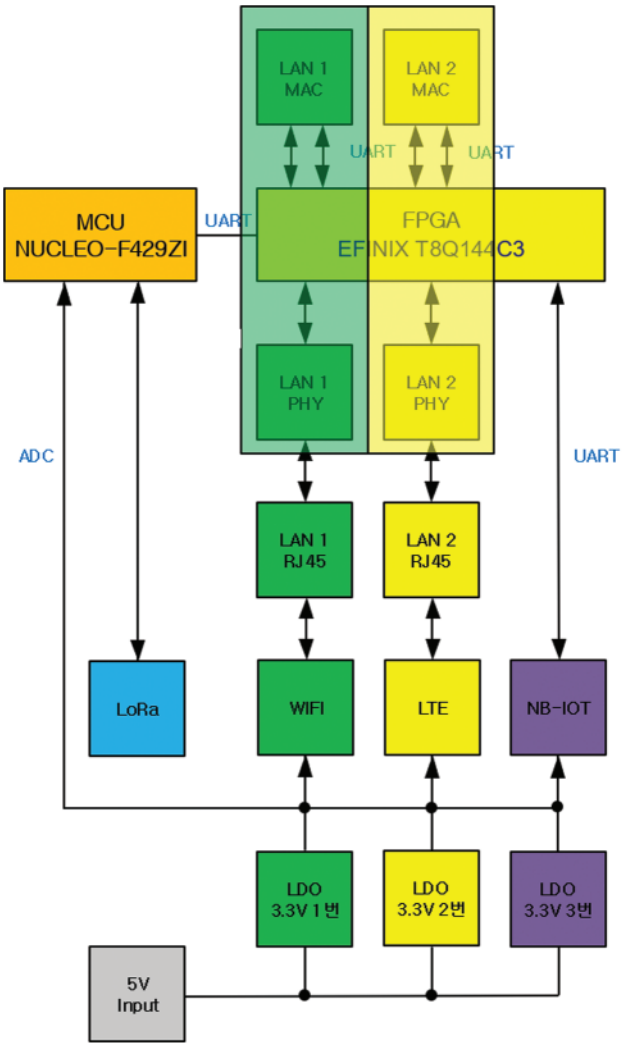

Figure 4: Communication signal fault detection

WiFi, LTE, and NB-IoT each make 3.3 Vs using separate Low Drop-Output (LDO)s and individual 3.3 Vs determine fault in real time using MCU ADCs. If a voltage of less than 3 V is detected due to a physical fault of the MAC and PHY chips, the MCU notifies the FPGA and the FPGA performs network switching according to priority as shown in Fig. 5.

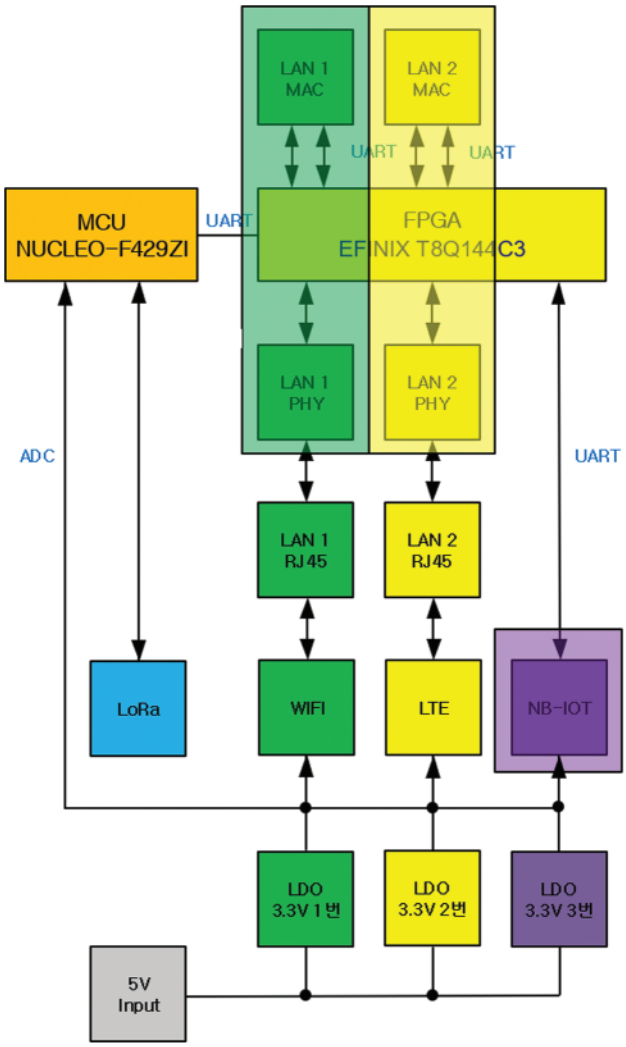

Figure 5: 3.3 V power fault detected by communication method

2.1.6 Operation Network Switch

- MCU, FPGA Information Exchange

The exchange of information such as fault status and operation network between MCU and FPGA is performed by communicating with each other using SPI (Serial Peripheral Interface) protocol in Fig. 6. Depending on the link failure, communication failure, power failure, operating network, etc., the FPGA internal registers are allocated as shown in Tab. 1, and the bit values by address are as shown in Tab. 2.

- Action by Switching Method

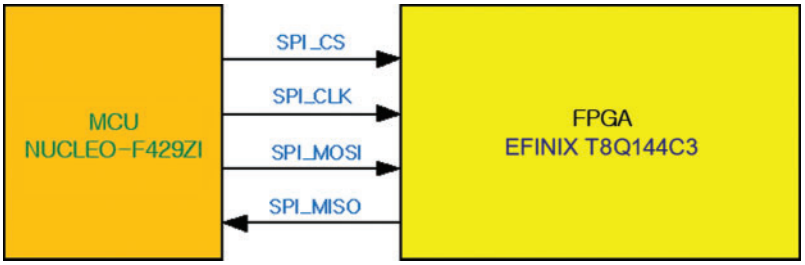

Figure 6: Configuration diagram of information exchange between MCU and FPGA

Table 1: FPGA internal register

Address	R/W	Register name	Function
0 × 01	R/W	Link Fault Detection	The MCU checks the link fault status and notifies the NB-IoT link fault.
0 × 02	Read	Communication Signal Fault Detection	The MCU checks the communication signal fault status.
0 × 03	Write	Power Fault Register	The power fault status is reported from the MCU to the FPGA.
0 × 04	R/W	Operation Network Register	Network currently in operation

Table 2: Address (0x01 ∼ 0x04)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	-	-	-	-	-	NB-IoT normal	LTE normal	WiFi normal
1	-	-	-	-	-	NB-IoT fault	LTE fault	WiFi fault

In the case of communication networks where fault is detected in FPGA, they are automatically switched by prioVrity. MCU periodically (100 milliseconds) checks the results determined by FPGA and notifies the server.

If the server algorithm requires switching the operation network, use the 0x04 operation network register. The contents of packets notified by the server are written in the FPGA internal register, and FPGA switches the operation communication network based on this. In this case, the switch to the operation network where the FPGA has detected a fault is not carried out.

In the case of the internal operation of the FPGA when switching the operation network, the RS-232 for communication connected to the MCU automatically connects to the corresponding communication device inside the FPGA, and the FPGA initializes the newly connected device. If the MCU is sending a message to the server, the MCU sends the packet back to the server when an operation network switch occurs.

2.2 Security in Software

The core of a network is to provide a variety of services seamlessly in a single network. In order to support various services, we are preparing various technologies for mobile communication such as NB-IoT, LTE, WiFi, and LoRa. The network part of our system transmits and receives 5 types of information (data, firmware, resource, device, network) confirmed from the control unit. Also, security in the network is an important issue. In addition to transmitting data encrypted (Advanced Encryption Standard Algorithm, AES) using edge computing in the device part, an intrusion prevention system was applied as a security problem in the process of transmission to the network server.

2.2.1 Intrusion Prevention System

Intrusion prevention systems are software applications that monitor for malicious activity or unauthorized access on a network [25]. IPS acts on suspicious packets and automatically halts execution if server takes unusual actions by detecting attack signatures and observing traffic on the network. This is necessary in edge computing to prevent attackers by passing edge servers to break into the main network and seizing information. IPS is largely divided into host-based and network-based systems. First, the technical characteristics of host-based IPS are largely separated by the way it works with the kernel to intercept and handle kernel events. The technical features of host-based IPS are largely distinguished by the way it works with the kernel to intercept kernel events and the way it operates independently of the kernel. The former can be classified as Trust Operating System products with access control capabilities, and the latter can be classified as products that filter events that violate certain rules using signatures and behavior-based analytics algorithms.

The technical features of network-based IPS consist of real-time packet processing, techniques to minimize misdetection, detection techniques of variant and misuse attacks, and real-time response techniques for each situation [26]. It is also essential to block malicious sessions through various kinds of prevention methods and methods (signature, anomaly detection on protocol) as a system capable of supporting session-based detection.

The biggest advantage of IPS is that it is an active security measure that can minimize attack damage by actively blocking attacks before an attack can cause damage. In addition, IPS provides higher security because it proactively complements vulnerabilities in OS or applications and can block worms or overflows, especially Anomaly traffic or unknown attacks.

We trained a machine learning approach to monitor malicious activity in real time using that model. The proposed system modeling includes 5 stages of data collection, preprocessing, feature selection, model selection, training and testing, and result evaluation. Fig. 7 below is a flowchart of the proposed intrusion prevention system.

2.2.2 Data Collection

The KDD data set is well known in intrusion detection technology research [27]. A lot of research is going on to improve intrusion detection, and the study of data used to train, and test detection models is a major issue because better data quality can improve offline intrusion detection. This data set is mainly used in intrusion prevention system. KDD dataset contains 41 features and class label that is normal or attack. It is more efficient in the method with detection rate than KDD by using the NSL-KDD data set in which duplicate items are removed from the KDD dataset [28]. The NSL-KDD data set includes KDDTrain+.ARFF for training and KDDTest+.ARFF

for testing. ARFF is used to test binary category problems. And KDDTrain+.TXT and KDDTest +.TXT, which contain 5 categories (normal and 4 types of attacks: Dos, Probe, R2L, U2R), are used to train and test the dataset, respectively.

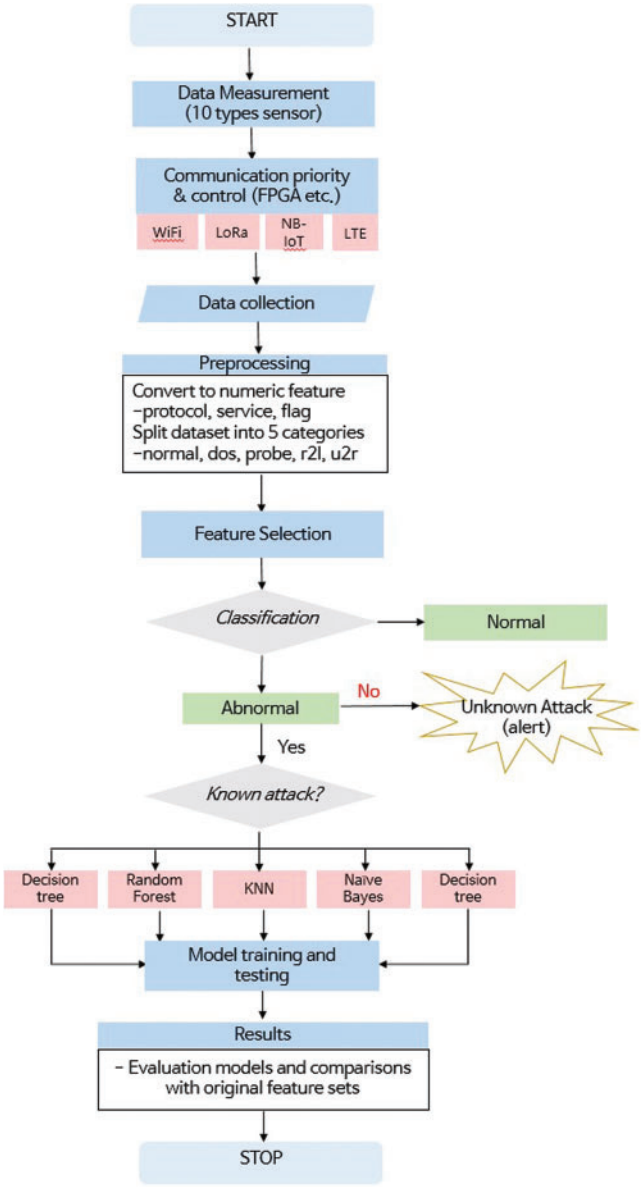

Figure 7: Proposed intrusion prevention system model

In the attack type, DoS is a denial-of-service attack, which depletes the victim's resources and makes it impossible to process legitimate requests. Relevant features include "source bytes" and "percent packets with errors". Probe is to obtain information about remote victims by surveillance and other investigation attacks. Relevant features are "duration" and "source byte". U2R is a type of attack that provides unauthorized access to local superuser (root) privileges. An attacker logs

in to the victim's system using a generic account and attempts to gain administrator privileges by exploiting some of the victim's vulnerabilities. Relevant features include "number of file creations" and "number of shell prompts called." R2L is a type of attack that gains unauthorized access to a remote system and local access to the victim system. Relevant features include "duration", "service requested", and "number of failed login attempts".

2.2.3 Preprocessing

The data set contains numeric and non-numeric features. Types with non-numeric properties such as protocol type, service, and flag must be converted to numeric features in preparation for the next step used as training and test input. It also must convert Normal and 4 types of attacks (DoS, Probe, R2L, U2R) belonging to the class characteristics into numeric types. This conversion is done in the preprocessing stage.

2.2.4 Feature Selection

Feature selection is important for both training and classification processes that effectively reduce the amount of data required for processing, memory and CPU usage. Among the 41 features included in the KDD dataset, it is necessary to exclude features that do not significantly affect intrusion prevention.

In addition, among the many studies that selected features related to the four attack groups included in the KDD dataset [29–31] selected by the voting system for features that were used most among these studies were referenced. Feature numbers 2, 4, 5, 21 were selected for all experiments, and added feature numbers 10, 14, 21, 22, 28, 36, 40, 41 for Feature 12 and 1, 4, 7, 8, 11, 16, 23, 24, 27, 29, 30 and 37 for feature 24. And 41 features were also used in the experiment to compare the execution time of machine learning. Tab. 3 shows the main features related to the for attack groups. Features not included in the attack type were deleted.

Table 3: Important features related to the four attack groups

No.	Feature paper	Kayacik et al. [29]	Tang et al. [30]	Olusola et al. [31]	Amiri et al. [28]	Zargari et al. [32]
1	duration	-	-	-	R2L	-
3	service	R2L	R2L	R2L	R2L	voted
4	flag	Probe	Probe	-	-	-
5	source_bytes	Normal, DoS, R2L, U2R	Normal, DoS, R2L, U2R	DoS, Probe	R2L	voted
6	destination_-bytes	DoS, R2L, U2R	DoS, R2L, U2R	R2L	DoS, R2L	voted
7	land	DoS	DoS	DoS		
10	hot	-	-	-	R2L	-
11	failed_logins	-	-	R2L	-	-
14	root_shell	-	-	U2R	U2R	-
16	num_root	U2R	U2R	-	-	-

Continued)

Table 3: Continued

No.	Feature paper	Kayacik et al. [29]	Tang et al. [30]	Olusola et al. [31]	Amiri et al. [28]	Zargari et al. [32]
21	is_hot_login	-	-	-	U2R	-
22	is_guest_-login	-	-	-	R2L	-
23	count	-	-	R2L	-	-
24	srv_count	-	-	U2R	-	-
27	rerror_rate	Probe	Probe	-	-	-
28	srv_rerror_-rate	-	-	Probe	Probe	-
29	same_srv_-rate	-	-	Normal	-	-
30	diff_srv_rate	DoS	DoS	Probe	-	-
34	dst_host_-same_src_-port_rate	-	-	Probe, U2R	DoS	
35	dst_host_-srv_diff_-host_rate	Probe	Probe	-	-	-
38	dst_host_-srv_serror_-rate	R2L	R2L	R2L	DoS	voted
40	dst_host_-rerror_rate	-	-	-	Probe	-
41	dst_host_-srv_rerror_-rate	-	-	-	Probe	-

2.2.5 Model Selection

The program first distinguishes Normal/Abnormal data using Multi Layer Perceptron (MLP), K-Nearest Neighbors (KNN), and Support Vector Machine (SVM) models. The models use training data consisting of normal traffic and known attack traffic. The next step is to distinguish between unknown and known attacks among abnormal data. Among machine learning techniques, we train using Decision Tree, Random Forest, KNN, Naïve Bayes, and SVM, which are well known as supervised learning.

MLP is called Feed-forward Neural Network, or Neural Network, and it is a generalized form of Linear Model that makes decisions through several steps. In the MLP, the process of creating the sum of weights is repeated several times. First, the hidden unit constituting the intermediate step is calculated, and the sum of weights is calculated again to calculate the final result using this. KNN is one of the simplest learning algorithms and classifies each data into one of k sets by measuring the distance of data in a short time. The training data is a vector in a multidimensional feature space, each with an item class name, and the training step of the algorithm is to store only the feature vector and item class name of the training sample. SVM is one of the fields of machine learning, a supervised learning model for pattern recognition and data analysis. And it

is mainly used for classification and regression analysis. Given a data set belonging to one of the two categories, the SVM algorithm creates a non-probabilistic binary linear classification model that determines which category the new data belongs to and is expressed as a boundary in the mapped space. Decision Tree (DT) is one of the most widely used classification algorithms in data mining. It operates in a split-and-conquer method, and recursively splits the training data set based on attributes until the stopping condition is satisfied. Each node has several corners that specify the possible values or ranges of values for the selected attribute in the node, specifying the attributes that best divide the DT's data set into classes. The most important issue when constructing DT is the value chosen to divide the tree node. Random Forest (RF) is an ensemble classifier used to improve accuracy. RF consists of many decision trees and has lower classification errors compared to other existing classification algorithms. The advantage of RF is that the generated forest can be saved for future reference, overcomes fitting problems, and is automatically generated in RF accuracy and variable importance. Naïve Bayes (NB) algorithm is an algorithm based on Bayes' theorem that can be used for classification data sets. This algorithm is based on a simple assumption that attribute values with conditions independent of the target variable are considered. NB provides a systematic method for the data analysis process with a probabilistic model.

2.2.6 Training and Testing

Run the training dataset as input to machine learning and use the test dataset to test the training model's ability to generalize. Tab. 4 describes the attack class and the number of patterns per class. The proposed algorithm is used to train the pattern selected as 125973 in the NSL-KDD data set and test the 22,544 patterns.

Table 4: Number of patterns by attack type

Training data set		Testing data set	
Class	Number of patterns	Class	Number of patterns
Normal	67,343	Normal	9,711
DoS	45,927	DoS	7,458
Probe	11,656	Probe	2,421
R2L	995	R2L	2,754
U2R	52	U2R	200
Total	125,973	Total	22,544

2.2.7 Model Evaluation

In evaluating an intrusion prevention system, there is an accuracy limit that is inappropriate to use only classification accuracy as an evaluation index. So, it must be applied together with several classification indexes to overcome. The Confusion Matrix is an index that shows how much confusion can occur when a learned classification model performs prediction as a method that is well used as a performance index in classification (Shown in Tab. 5). In order to objectively evaluate the predictive and generalization ability of the model, the precision, recall, False Positive

Rate (FPR), accuracy, and F1 score of abnormal behavior detection are used as evaluation indicators.

Table 5: Confusion matrix

Attack		Actual class		Measure
		YES (True)	NO (False)	
Predicted class	Yes (positive)	TP or DR	FP	Precision TP/(TP*FP)
	No (Negative)	FN	TN	-
Measure		Recall, Sensitivity TP/(TP*FN)	Specificity, True Negative Rate = TN/(TN*FP)	Accuracy = (TP*TN)/(TP*TN* FP*FN)

Precision is the probability that the model's prediction result is positive, and the actual value is positive. It is also called Positive Predictive Value (PPV) as an indicator of how well you match your positives.

2.3 Machine Learning

The process of predicting the cyberattack is used not only for securing reliability of the existing data and defining the cause of faults that had already occurred also in forecasting future to detect the user's risk in advance [33]. This is a list of the machine learning classifiers used in this experiment to increase the efficiency of detection of network anomalies and attacks [34]: SVM, KNN, MLP, Naïve Bayes and Random Forest. SVM technique is well-developed supervised learning model for pattern recognition and data analysis as one of the fields of machine learning, and mainly used for classification and regression analysis [35,36] proposed a DDoS anomaly and attack detection technique based on SVM. This scheme shows that values such as source IP, source port and flow item speed, flow packet standard deviation, flow byte deviation, and pair ratio are extracted from the switch flow table of the SDN architecture related to DDoS attacks. Network DoS attack and malicious code detection [37] and network intrusion [38] can be confirmed by using KNN technique [39,40]. The experiment was conducted by classifying the network state into two (attack or normal) or three (DDoS source, victim, normal) using the MLP technique [41]. This makes it possible to classify attacks with a high true positive rate while keeping the false positive rate low. The study to confirm the presence or absence of intrusion prevention was conducted by introducing three well-known classification techniques, MLP, NB and Random Forest, among which MLP detected invasion with the highest accuracy [42]. Random Forest is a machine learning classifier made up of a number of decision trees that operate as a group, where the most voted prediction is accepted [43]. While the Multi Class Classifier and Random Forest algorithms detected 100% of all web-based attacks, the Naïve Bayes and Naïve Bayes Updatable algorithms detected only HTTP Flood among the four attacks, and a 96% rate was detected [44].

3 Results

3.1 Fault Detection Using FPGA

Using the fabricated FPGA board, we checked the operation in the situation of link fault detection, communication signal fault detection, and power fault detection. We implemented FPGA by synthesizing EFINIX TRION T8Q144C3 FPGA using EFINIX Efinity 2020.1 tools and TRION T8Q144C3. And the Tab. 6 below is the description of each input/output of the FPGA.

Table 6: Description of FPGA

Name	Description
clk	FPGA internal reference clk (40nsec)
mac_rxclk	Communication clk to the upper MAC of the FPGA
mac_rxd	Communication data to the upper MAC of the FPGA
wifi_rxclk	Communication input from WiFi in FPGA
wifi_rxd	Communication data input from WiFi in FPGA
lte_rxclk	Communication input from LTE in FPGA
lte_rxd	Communication data input from LTE in FPGA
spi_clk, spi_cs, spi_mosi	Signal from MCU to communicate with FPGA
wifi_link_fail, lte_link_fail, nb_iot_link_fail	External communication device cable or device error
wifi_comm_fail, lte_comm_fail	Communication signal error
reg_5_out	FPGA internal registers (display the operating device) - 00000001: WiFi - 00000010: LTE - 00000100: NB-IoT - 00001000: LoRa

Fig. 8a is a normal state and WiFi with high priority is basically connected. In the case of link fault detection as shown in Fig. 8b, when a link failure occurs for each communication, the process of switching after detecting the relevant situation is shown. First, as the WiFi_link_-fail signal fell from high to low, the abnormal state is detected, and then the communication is switched from WiFi to LTE. When an abnormal state is detected for each communication, it is connected to the next communication according to the priority. And the WiFi_link_fail signal returns to normal, the communication from LoRa to WiFi is restored.

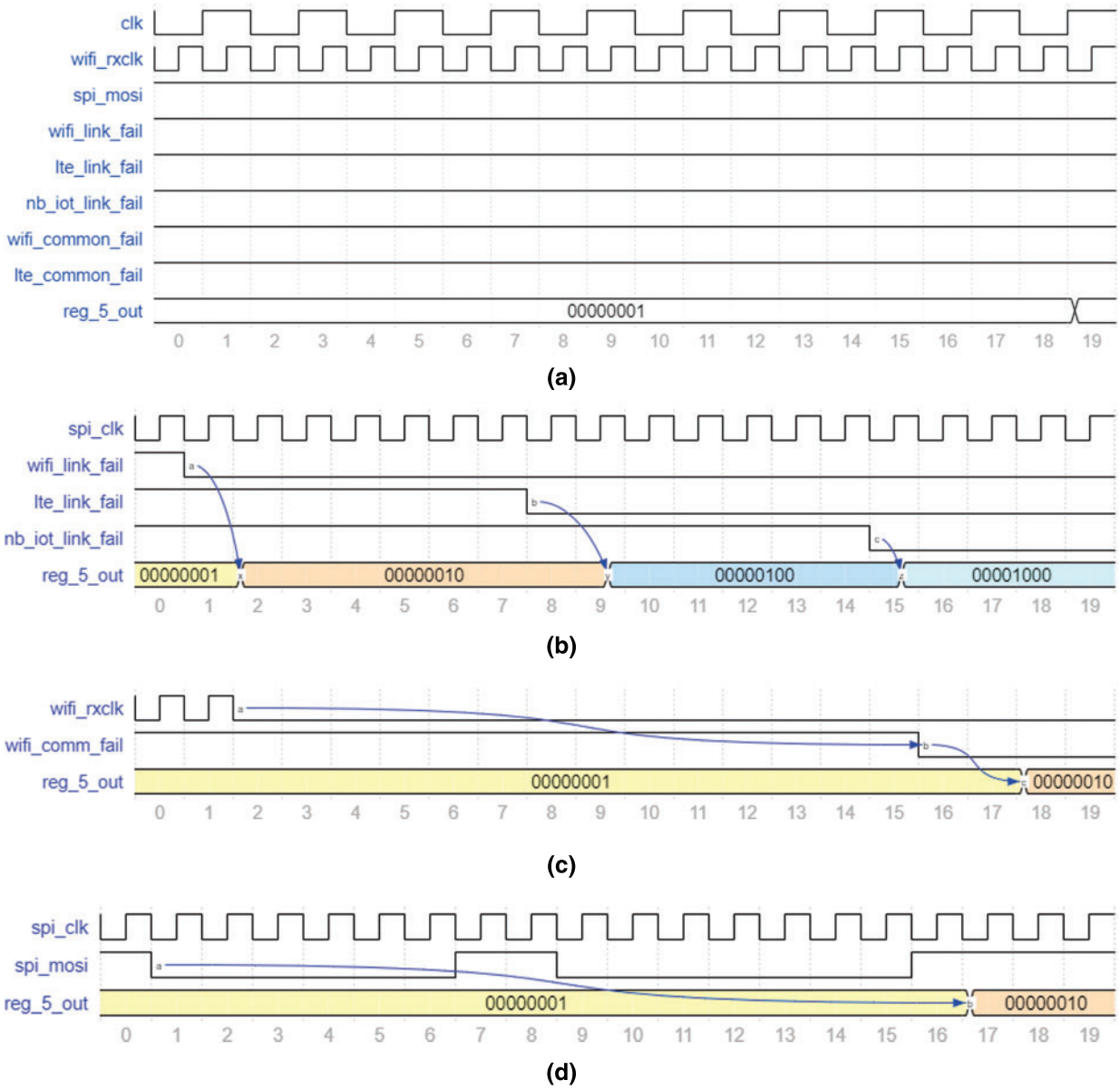

Figure 8: Timing diagram for each fault detection: (a) Normal state, (b) Link fault detection, (c) Communication signal fault detection, (d) Power fault detection

For communication signal fault detection, the process of switching communication from WiFi to LTE after detecting an abnormality in the WiFi communication signal through the WiFi_rxclk signal in the FPGA is shown in Fig. 8c. When a power fault is detected as shown in Fig. 8d, the MCU detects a WiFi power error and informs the FPGA through SPI communication. And the signal is transferred from WiFi to LTE by writing the communication detected for power failure in the FPGA internal register 0x03 indicating power fault detection.

3.2 Autonomous Defense System Using Machine Learning

It is an excellent intrusion prevention system with high accuracy and detection speed but low false positive rate. The false alarm rate is proportional to the misclassification rate.

First, we used the SVM, MLP, and KNN methods to classify them into Normal and Abnormal data. Classification for the three models using NSL-KDD data shows an accuracy of 91% in KNN, 85% in MLP, and 52% in SVM. In the case of SVM, the accuracy differs significantly depending on the kernel types. RBF was used because the data set used in this study did not match linear. The parameters used at this time are C and gamma, and the values are 1.0 and 0.1. Because the misclassification was not strictly managed by giving a large C value, the model shows a relatively low prediction rate.

In the dataset classified as abnormal data, the attack types that are not included in the training are compared. In case of an unknown type, an alarm is given. And in case of a known attack, five machine learning methods are trained and tested, and the predicted results are output. However, in trivial classes such as U2R attacks, especially in the case of DT and RF, the result of 4 features shows a higher F1 score than when 8 is selected. These methods show that the classification is poor due to the data characteristics of a larger number of features added to the classification through pruning. The reason why the NB method yields a value of 0 is that this method determines the classification in a relatively simple method compared to other models. This method can be used to judge features of low importance and high features equally, leading to results that are not suitable for the model.

Fig. 9 show the execution time of the five models. The amount of time change according to the number of features is the smaller the feature in the order of DT, NB, and SVM, the more effectively the time was shortened. In comparison by model, the execution time was shortened in the order of NB, DT, RF, and KNN, and the execution time of SVM was the longest which is plotted as a log scale graph. SVM takes a lot of time when processing large datasets. In the SVM method, it can avoid wasting time resources by selecting fewer features.

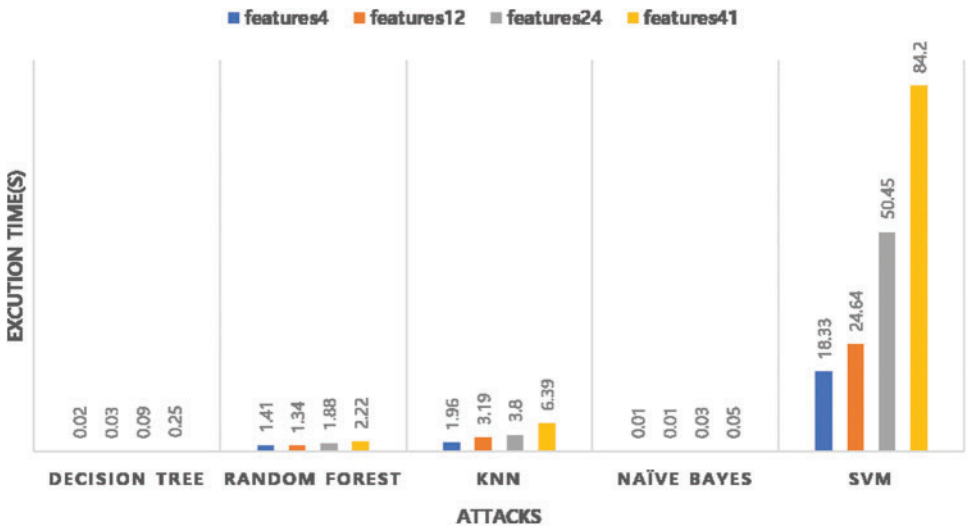

Figure 9: Executions time by ML model

The Receiver Operating Characteristics (ROC) curve which draws the relationship between False Positive Rate (FPR) and True Positive Rate (TPR). (For FPR), as shown in Fig. 10 and Tab. 7. The ROC curve has the advantage which is not sensitive to the class distribution. Also, performance can be compared with the Area Under Curve (AUC) value obtained from the ROC curve. AUC has a value from 0 to 1, and the larger the better.

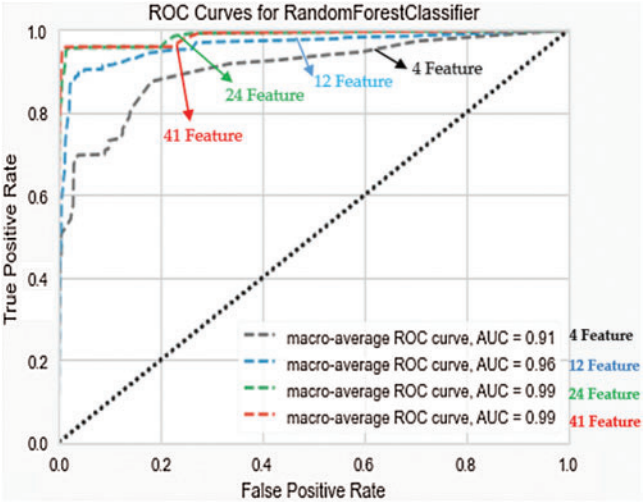

Figure 10: ROC curve with random forest

Tab. 8 shows the values of FPR, and TPR for dos, probe, u2r, and r2 l predicted by five machine learning techniques as shown in Fig. 10. We classified the TPR/FPR that can be extracted from the actual network attack situation using the KDD Dataset. And we confirmed that the IPS can detect and block external intrusion.

4 Discussion

We build edge computing system and control it with FPGA, MCU to prioritize four communication methods. This configured an edge system in the sensing part, reducing the load on the server and ensuring communication stability. The data collected to the server was considered by performing end-to-end encryption, and Autonomous defense systems were modeled on the server using machine learning to target the server's incoming and outgoing logs. Server logs leverage existing intrusion prevention datasets, which need to be used in real-world environments.

Machine learning modeling for intrusion anomaly detection showed relatively good results in RF, but poor performance in various other algorithms. To improve this, future studies need to consider more accurate properties and counts in feature selection, and the appropriate amount of data and parametric coefficient adjustments are required for SVM models in machine learning. Furthermore, KNN is an effective algorithm when the number of learning data is high, but it

is unclear how many optimal neighbors (k) and which distance scales are suitable for analysis, requiring selection for each characteristic of the data.

In this paper, we have seen many possibilities that can be effective in detecting anomalies and preventing intrusion through various machine learning methods. To specify the method, run the Data set in the future by bootstrap to add content. And it is necessary to increase accuracy by using Stratified k-fold-cross-validation. So, we need to conduct future experiments to coordinate this hybrid method for better accuracy. We then have to experiment with contextual optimization algorithms by collecting data from live networks.

Table 7: AUC with (a) DT, (b) KNN, (c) SVM, (d) NB

(a) Decision tree	
Features	**AUC**
4	0.78
12	0.81
24	0.83
41	0.85
(b) KNN	
Features	**AUC**
4	0.83
12	0.86
24	0.85
41	0.83
(c) SVM	
Features	**AUC**
4	0.78
12	0.78
24	0.77
41	0.78
(d) Naïve bayes	
Features	**AUC**
4	0.64
12	0.64
24	0.74
41	0.74

Table 8: TPR and FPR value of four class classifications with 41 features. (a) Random forest, (b) KNN, (c) Decision tree, (d) Naïve bayes, (e) SVM

(a) Random forest		
Attack type	TPR	FPR
dos	99.43	0.12
probe	40.75	0.17
u2r	54.05	0.11
r2l	100	16.64
(b) KNN		
Attack type	TPR	FPR
dos	96.46	17.59
probe	5.82	0.24
u2r	35.14	0.16
r2l	83.27	23.35
(c) Decision tree		
Attack type	TPR	FPR
dos	99.4	0.30
robe	34.51	0.05
u2r	56.76	0.42
r2l	99.91	13.88
(d) Naïve bayes		
Attack type	TPR	FPR
dos	93.94	92.7
probe	10.23	5.12
u2r	29.73	0.02
r2l	0	0
(e) SVM		
Attack type	TPR	FPR
dos	100	100
probe	0	0
u2r	0	0
r2l	0	0

5 Conclusion

With the advent of the 5G era, we designed and implemented a mobile IoT edge computing system and studied how to enhance security in hardware and software. For this, we have secured the stability and security of communication through the priority of four communication methods through FPGA and MCU. Server logs were utilized to model intrusion prevention systems to

identify behaviors that violate security policies to detect intrusions early. This autonomous defense system performs control over edge computing and predicts abnormal intrusions in the server side as well.

This is due to the importance of network security in this system. Leveraging the NSL-KDD dataset, we test by feature count using five machine learning methods to avoid wasting resources and find models suitable for attack detection and classification. As a result, the selection of features that are highly relevant to intrusion prevention characteristics is important. The performance of classification algorithms according to their characteristics showed that RF performed best on the ROC curve and SVM were the worst in terms of execution time. We also used a learning algorithm by creating a dataset targeting server logs, which we confirm can detect/control external attacks and build an autonomous defense system through this way.

Acknowledgement: This work was supported Korea Environmental Industry & Technology Institute (KEITI) grant funded by the Korea government (Ministry of Environment). Project No. RE202101551, the development of IoT-based technology for collecting and managing Big data on environmental hazards and health effects.

Funding Statement: This research was funded by Korea Environmental Industry & Technology Institute (KEITI), Grant Number RE202101551and The APC was funded by Ministry of Environment (ME).

Conflicts of Interest: The author declares that they have no conflicts of interest to report regarding the present study.

References

[1] C.-H. Chen, M.-Y. Lin and C.-C. Liu, "Edge computing gateway of the industrial internet of things using multiple collaborative microcontrollers," *IEEE Network*, vol. 32, no. 1, pp. 24–32, 2018.

[2] S. Tu, M. Waqas, S. U. Rehman, M. Aamir, O. U. Rehman *et al.*, "Security in fog computing: A novel technique to tackle an impersonation attack," *IEEE Access*, vol. 6, pp. 74993–75001, 2018.

[3] S. Tu, M. Waqas, Y. Meng, S. U. Rehman, I. Ahmad *et al.*, "Mobile fog computing security: A user-oriented smart attack defense strategy based on DQL," *Computer Communications*, vol. 160, pp. 790–798, 2020.

[4] J. Wan, M. Waqas, S. Tu, S. M. Hussain, A. Shah *et al.*, "An efficient impersonation attack detection method in fog computing," Computers, *Materials & Continua*, vol. 68, no. 1, pp. 267–281, 2021.

[5] W. Shi, J. Cao, Q. Zhang, Y. Li and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[6] M. Haus, M. Waqas, A. Y. Ding, Y. Li, S. Tarkoma *et al.*, "Security and privacy in device-to-device (D2D) communication: A review," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 1054–1079, 2017.

[7] P. Radanliev, D. D. Roure, M. V. Kleek, U. Ani, P. Burnap *et al.,* "Dynamic real-time risk analytics of uncontrollable states in complex internet of things systems: Cyber risk at the edge," *Environment Systems and Decisions*, vol. 41, pp. 236–247, 2021.

[8] M. Tanveer, G. Abbas, Z. H. Abbas, M. Waqas, F. Muhammad *et al.*, "S6AE: Securing 6LoWPAN using authenticated encryption scheme," *Sensors*, vol. 20, no. 9, 2707, pp. 1–23, 2020.

[9] M. Waqas, S. Tu, S. U. Rehman, Z. Halim, S. Anwar *et al.*, "Authentication of vehicles and road side units in intelligent transportation system," *Computers, Materials & Continua*, vol. 64, no. 1, pp. 359–371, 2020.

[10] S. Tu, M. Waqas, S. U. Rehman, T. Mir, G. Abbas *et al.*, "Reinforcement learning assisted impersonation attack detection in device-to-device communications," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 2, pp. 1474–1479, 2021.
[11] J. Zhang and G. Qu, "Recent attacks and defenses on FPGA-based systems," *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, vol. 12, no. 3, pp. 1–24, 2019.
[12] R. Roman, J. Lopez and M. Mambo, "Mobile edge computing, fog *et al.*: A survey and analysis of security threats and challenges," *Future Generation Computer Systems*, vol. 78, pp. 680–698, 2018.
[13] M. Xiong, Y. Li, L. Gu, S. Pan, D. Zeng *et al.*, "Reinforcement learning empowered IDPS for vehicular networks in edge computing," *IEEE Network*, vol. 34, no. 3, pp. 57–63, 2020.
[14] H. Zimmermann, "OSI reference model-the ISO model of architecture for open systems interconnection," *IEEE Trans-Actions on Communications*, vol. 28, no. 4, pp. 425–432, 1980.
[15] S. Biookaghazadeh, M. Zhao and F. Ren, "Are FPGAs suitable for edge computing?," in *Proc. USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18)*, Boston, MA, 2018.
[16] S.-S. Kim and S. Jung, "Hardware implementation of a neural network controller with an MCU and an FPGA for nonlinear systems," *International Journal of Control, Automation, and Systems*, vol. 4, no. 5, pp. 567–574, 2006.
[17] Y. Alkabani and F. Koushanfar, "Designer's hardware trojan horse," in *Proc. IEEE Int. Workshop on Hardware-Oriented Security and Trust*, Anaheim, CA, USA, pp. 82–83, 2008.
[18] A. Waksman, M. Suozzo and S. Sethumadhavan, "FANCI: Identification of stealthy malicious logic using boolean functional analysis," in *Proc. the 2013 ACM SIGSAC Conference on Computer & Communications Security*, Berlin, Germany, pp. 697–708, 2013.
[19] S. M. Trimberger and J. J. Moore, "FPGA security: Motivations, features, and applications," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1248–1265, 2014.
[20] J. Zhang and G. Qu, "A survey on security and trust of FPGA-based systems," in *Proc. 2014 Int. Conf. on Field-Programmable Technology (FPT)*, Shanghai, China, pp. 147–152, 2014.
[21] P. Kocher, J. Jaffe and B. Jun, "Differential power analysis," in *Proc. Annual International Cryptology Conference*, Santa Barbara, CA, USA, Springer, pp. 388–397, 1999.
[22] M. Waqas, M. Ahmed, Y. Li, D. Jin and S. Chen, "Social-aware secret key generation for secure device-to-device communication via trusted and non-trusted relays," *IEEE Transactions on Wireless Communications*, vol. 17, no. 6, pp. 3918–3930, 2018.
[23] M. Waqas, M. Ahmed, J. Zhang and Y. Li, "Confidential information ensurance through physical layer security in device-to-device communication," in *Proc. 2018 IEEE Global Communications Conf. (GLOBECOM)*, Abu Dhabi, United Arab Emirates, pp. 1–7, 2018.
[24] M. Zhao and G. E. Suh, "FPGA-Based remote power side-channel attacks," in *2018 IEEE Symp. on Security and Privacy (SP)*, San Francisco, CA, USA, pp. 229–244, 2018.
[25] X. Zhang, C. Li and W. Zheng, "Intrusion prevention system design," in *Proc. the Fourth Int. Conf. on Computer and Information Technology*, Washington, DC, USA, pp. 386–390, 2004.
[26] A. Fuchsberger, "Intrusion detection systems and intrusion prevention systems," *Information Security Technical Report*, vol. 10, no. 3, pp. 134–139, 2005.
[27] P. Aggarwal and S. K. Sharma, "Analysis of KDD dataset attributes-class wise for intrusion detection," *Procedia Computer Science*, vol. 57, pp. 842–851, 2015.
[28] F. Amiri, M. Yousefi, C. Lucas, A. Shakery and N. Yazdani, "Mutual information-based feature selection for intrusion detection systems," *Journal of Network and Computer Applications*, vol. 34, pp. 1184–1199, 2011.
[29] H. G. Kayacik, A. N. Zincir-Heywood and M. I. Heywood, "Selecting features for intrusion detection: A feature relevance analysis on KDD 99 intrusion detection datasets," in *Proc the Third Annual Conference on Privacy, Security and Trust, New Brunswick*, Canada, vol. 94: Citeseer, pp. 1723–1722, 2005.
[30] P. Tang, R.-A. Jiang and M. Zhao, "Feature selection and design of intrusion detection system based on k-means and triangle area support vector machine," in *Proc. 2010 Second Int. Conf. on Future Networks*, Washington, D.C, USA, pp. 144–148, 2010.

[31] A. A. Olusola, A. S. Oladele and D. O. Abosede, "Analysis of KDD'99 intrusion detection dataset for selection of relevance features," in *Proc. the World Congress on Engineering and Computer Science*, San Francisco, CA, USA, vol. 1, pp. 20–22, 2010.

[32] S. Zargari and D. Voorhis, "Feature selection in the corrected KDD-dataset," in *Proc. 2012 Third Int. Conf. on Emerging Intelligent Data and Web Technologies*, Bucharest, Romania, pp. 174–180, 2012.

[33] J. H. Cho and H. Lee, "Optimization of machine learning in various situations using ICT-based TVOC sensors," *Micromachines*, vol. 11, no. 12, pp. 1092–1105, 2020.

[34] I. Tariq, M. A. Sindhu, R. A. Abbasi, A. S. Khattak, O. Maqbool *et al.*, "Resolving cross-site scripting attacks through genetic algorithm and reinforcement learning," *Expert Systems with Applications*, vol. 168, pp. 1–15, 2020.

[35] B. S. Bhati and C. Rai, "Analysis of support vector machine-based intrusion detection techniques," *Arabian Journal for Science and Engineering*, vol. 45, no. 4, pp. 2371–2383, 2020.

[36] J. Ye, X. Cheng, J. Zhu, L. Feng and L. Song, "A DDoS attack detection method based on SVM in software defined network," *Security and Communication Networks*, vol. 2018, pp. 1–8, 2018.

[37] J. W. Branch, C. Giannella, B. Szymanski, R. Wolff and H. Kargupta, "In-network outlier detection in wireless sensor networks," *Knowledge and Information Systems*, vol. 34, no. 1, pp. 23–54, 2013.

[38] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2015.

[39] M. Bagaa, T. Taleb, J. B. Bernabe and A. Skarmeta, "A machine learning security framework for IoT systems," *IEEE Access*, vol. 8, pp. 114066–114077, 2020.

[40] J. H. Cho, "Detection of smoking in indoor environment using machine learning," *Applied Sciences*, vol. 10, no. 24, pp. 8912–8929, 2020.

[41] C. Siaterlis and V. Maglaris, "Detecting incoming and outgoing DDoS attacks at the edge using a single set of network characteristics," in *Proc. 10th IEEE Symposium on Computers and Communications (ISCC'05)*, Murcia, Spain, pp. 469–475, 2005.

[42] M. Alkasassbeh, G. Al-Naymat, A. Hassanat and M. Almseidin, "Detecting distributed denial of service attacks using data mining techniques," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 1, pp. 436–445, 2016.

[43] K. E. Ismail, M. A. AbouRizka and F. A. Maghraby, "Machine learning model for multiclass lesion diagnoses," in *Proc. 2020 2nd Novel Intelligent and Leading Emerging Sciences Conf. (NILES)*, Giza, Egypt, pp. 397–402, 2020.

[44] M. Abushwereb, M. Mustafa, M. Al-Kasassbeh and M. Qasaimeh, "Attack based DoS attack detection using multiple classifier," 2020. [Online]. Available: https://arxiv.org/abs/2001.05707.