applied sciences

MDPI

Article

AI-Aided Proximity Detection and Location-Dependent Authentication on Mobile-Based Digital Twin Networks: A Case Study of Door Materials

Woojin Park [1], Hyeyoung An [1], Yongbin Yim [2,*] and Soochang Park [1]

[1] Department of Computer Engineering, Chungbuk National University, Cheongju 28644, Republic of Korea; woojin415@chungbuk.ac.kr (W.P.); elinyoung@chungbuk.ac.kr (H.A.); cewinter@chungbuk.ac.kr (S.P.)
[2] Agency for Defense Development (ADD), Daejeon 34060, Republic of Korea
[*] Correspondence: ybyim.cclab@gmail.com; Tel.: +82-10-5034-5575

Abstract: Nowadays, mobile–mobile interaction is becoming a fundamental methodology for human–human networking services since mobile devices are the most common interfacing equipment for recent smart services such as food delivery, e-commerce, ride-hailing, etc. Unlike legacy ways of human interaction, on-site and in-person mutual recognition between a service provider and a client in mobile–mobile interaction is not trivial. This is because of not only the avoidance of face-to-face communication due to safety and health concerns but also the difficulty of matching up the online user using mobiles with the real person in the physical world. So, a novel mutual recognition scheme for mobile–mobile interaction is highly necessary. This paper comes up with a novel cyber-physical secure communication scheme relying on the digital twin paradigm. The proposed scheme designs the digital twin networking architecture on which real-world users form digital twins as their own online abstraction, and the digital twins authenticate each other for a smart service interaction. Thus, inter-twin communication (ITC) could support secure mutual recognition in mobile–mobile interaction. Such cyber-physical authentication (CPA) with the ITC is built on the dynamic BLE beaconing scheme with accurate proximity detection and dynamic identifier (ID) allocation. To achieve high accuracy in proximity detection, the proposed scheme is conducted using a wide variety of data pre-processing algorithms, machine learning technologies, and ensemble techniques. A location-dependent ID exploited in the CPA is dynamically generated by the physical user for their own digital twin per each mobile service.

Keywords: machine learning; digital twin; proximity detection; mutual authentication

Citation: Park, W.; An, H.; Yim, Y.; Park, S. AI-Aided Proximity Detection and Location-Dependent Authentication on Mobile-Based Digital Twin Networks: A Case Study of Door Materials. *Appl. Sci.* **2024**, *14*, 9402. https://doi.org/10.3390/app14209402

Academic Editors: Ryan Gibson and Hadi Larijani

Received: 13 September 2024
Revised: 9 October 2024
Accepted: 10 October 2024
Published: 15 October 2024

1. Introduction

Smart mobile devices are becoming the most common tool in everyday life. The performance and functionalities of such mobile devices have also highly improved. Various smart systems based on mobile devices such as smart parking [1,2], navigation [3,4], and rescue [5,6] have been come up with accordingly. In particular, smart mobile services based on mobile–mobile interaction, e.g., food delivery, e-commerce, ride-hailing, and so on, have grown rapidly. This is because mobile devices have the advantage of providing online interaction easily through mobile applications.

Matching users who interact with each other is an important process in a digital service. In the case of the food delivery service, a user who requests food is matched with a delivery man. In the case of ride-hailing services, a user who reserves a car is matched with a driver. In this way, the user can be matched with someone using the digital service. However, the digital service does not guarantee that the user finds a matching person in the physical world. Although the digital service provides information about the matching person, the responsibility for finding the matching person in the physical world falls on the user. It creates a situation where the user should physically contact an unidentified someone to check whether someone matches the user.

Before the user contacts an unidentified person to find the matching person, it is hard for the user to know whether the person is a criminal, a patient with an infectious disease, or a matching person. In this situation, some problems could occur when the user physically contacts the person [7]. Let us assume the user waits for the package they ordered. Someone rings the bell of his house. The person who rings the bell says that they have a package. The user then opens the door to receive the package. Unfortunately, the person who rang the bell was a criminal deceiving as a delivery man. In the end, the user is put in danger. There are also health concerns due to infectious diseases such as COVID-19. When people communicate face to face to interact, infection may occur through contact. In this way, physical contact with the unidentified person cannot ensure the user's safety.

Digital twin architecture can be a solution to resolve problems that may arise from interaction in the physical world. The characteristic of the digital twin is that activities in the physical world are also performed in the digital world because the digital twin is an online abstraction that reflects physical objects. Various physical objects are transformed into digital twins such as city [8], vehicle [9], robot [10], and so on. So, it can be said that interaction between digital twins equals interaction between physical objects. A method of representing people as digital twins is also used [11]. Instead of unsafe interactions in the physical world, users can interact without physical contact by using the digital twin. Consequently, the system relying on digital twin architecture can provide secure interaction to the user.

In interaction between the users, mutual recognition is required to find the other person. Mutual recognition should provide functions that allow users to confirm the identity of the matching person they want to interact with and ensure that the matching person is within the interaction range to interact. In mutual recognition, proximity detection is used to check the interaction range.

This paper proposes a novel smart interaction system using digital twin architecture to prevent physical interaction. The contributions of the paper are as follows:

1. To convert physical interaction to digital interaction, the system uses digital twin architecture. Based on representing physical objects to digital twins, the system provides physically contactless interaction to the users.
2. Dynamic beaconing and proximity detection allow the users to find the matching user in the physical world. Also, we improve the performance of proximity detection using various pre-processing, machine learning models, and ensemble techniques.
3. The users match the matching users in the physical world with their digital twins using location-dependent IDs. The location-dependent IDs contain the user's location and time information so that it ensures that the user who has the specific digital twin is on site and in real time.
4. To sum up, the system provides secure interaction to the users without physical contact. It can prevent the problems that occur in the physical contact situation. Cyber-physical authentication (CPA) is offered with proximity detection, dynamic beaconing, location-dependent ID, dynamic ID allocation, and digital twin architecture for interaction.

The paper is organized as follows. In Section 2, the background about authentication and distance measurement schemes is introduced. In Section 3, the explanations of the proposed system and the environment for the proximity detection experiment are described. In Section 4, pre-processing and ensemble techniques that are used to improve the accuracy of proximity detection are introduced. In Section 5, the results of the proximity detection experiment are shown. In Section 6, an analysis of the results is described. Lastly, Section 7 concludes the paper.

2. Background of Analysis

2.1. Authentication

Authentication to confirm the person's identity is carried out in various ways. Authentication can be classified into three types: Physical authentication (PA), Cyber authentication (CA), and Cyber-Physical authentication (CPA). PA is the most common method to authenticate identity. For example, when people enter the theater to watch a movie, they should buy tickets and show them to a clerk. The clerk identifies the people to check whether they have a valid identity on site. The characteristic of PA is that authentication is carried out on site and in person. This makes it difficult for remote malicious users to interfere with authentication. However, PA must require physical contact with an unidentified person. When the user contacts the unidentified person to check their identity, the user does not know whether the unidentified person is a criminal, a patient with an infectious disease, or a normal person. This leads to a situation where the user's safety is at risk from physical contact.

Cyber authentication (CA) is another type of authentication. CA is carried out in the digital world. For example, the digital signature is used to prove the author of a digital document [12,13]. Digital ID is a common method to prove the user's identity to the system [14,15]. Blockchain is also used to enhance the integrity of data in the authentication process [16,17]. The characteristic of CA is that authentication does not require physical contact. So, there is no risk of problems from physical contact. However, there is a risk that a remote malicious user can interfere with the authentication.

Cyber-physical authentication (CPA) is a novel authentication that combines PA and CA. It authenticates the identity with physical and digital information. Amazon Key is an example of the CPA [18]. It provides authentication for a delivery man. When the delivery person arrives at a customer's house, they request the Amazon system to open the door. Then, the delivery person's identity is authenticated by Amazon. If authentication is successful, the door opens. This service uses the physical environment of the delivery person's location and the digital process using the Amazon system. There is another CPA [19]. The authors propose a delivery authentication system using a bar-code and Kerberos authentication. In this system, the users must meet to check the bar code. After that, the authentication using Kerberos is performed. In [20], users authenticate their identity using digital information such as a username and password. Then, continuous authentication is performed using RSSI obtained from the physical world.

The proposed system uses a digital twin to provide safety interaction. The system takes physical information, such as a proximity level between users, and digital information, such as a dynamic ID. Also, the dynamic ID, which is used for authentication, is generated based on the user's location. Because physical information and digital information are used for authentication, it can be seen that CPA is applied to the system. Table 1 shows the authentication types described.

Table 1. Three authentication classifications depending on the environment.

Authentication Type	Technology	Description
Physical Authentication (PA)	Tickets Identity Document	The authentication process takes place only in the physical world.
Cyber Authentication (CA)	Digital Signature [12,13] Digital ID [14,15] Blockchain [16,17]	The authentication process takes place only in the digital world.
Cyber-Physical Authentication (CPA)	Amazon Key [18] Research in [19,20]	The authentication process takes place in the physical world and the digital world.

2.2. Distance Estimation

Distance estimation is one of the methods that measure the distance between the users to find the user's location. Location-based systems (LBS) are widely used to find and utilize a user's location. There are various data sources used for LBS such as LiDAR [21,22], sonar [23,24], and vision [25]. Additionally, wireless signals are popular methods to find location [26–28]. In [26], authors introduce the localization method using WiFi in an apartment environment. They use the RSS value of Wi-Fi and multiply it by a manually set weight to reduce the time variance of the RSS. In [27], authors use the BLE signal and fingerprinting approach. A denoising autoencoder is adopted to improve localization performance. Wi-Fi and BLE have the advantage that they can be used indoors as well as outdoors. In [28], BLE is better suited to indicate the location in more detail than Wi-Fi. The Kalman filter is used to remove noise in RSS. Even if LBS is a valuable method to find the user's location, LBS needs environments with devices that provide indirect information about location, such as Wi-Fi APs and BLE beacons. Also, the data for making LBS are location-dependent so that LBS is sensitive to the environment.

Another method to measure the distance is a proximity-based system (PBS) that measures and utilizes the proximity between the users. Wireless signals such as Wi-Fi [29,30] and BLE [31–34] are often used to measure proximity between users. In [29], a method to check whether two mobile devices that can use Wi-Fi are close to each other, within 2 m, is proposed. In [30], the authors use Wi-Fi and BLE signals to estimate the proximity between two users. In [31], proximity is used to select the beacon that is nearest to the scanner. They demonstrate that the density of beacons can affect the accuracy of proximity detection. In [32], Bayesian filtering is used to improve proximity accuracy. Beacons are used in their experiments. However, using a beacon is not suitable for interaction between people. This is because people rarely own beacons in their daily lives. In [33], the authors use the Kalman filter to reduce signal fluctuation. Then, proximity detection is performed using pre-processed RSSI data.

The aforementioned PBS provides distance between the beacon and the user. However, most studies use static beaconing that the position of the beacons is fixed. Because the position of the beacon is fixed, deployment of the system is not flexible. Also, there is no consideration for obstacles that interrupt physical interaction between users. It indicates that the users should be in physical contact to use the system. So, we propose a novel proximity detection that includes dynamic beaconing and consideration of obstacles that prevent physical contact. In Table 2, the comparison of recent studies and our system is represented.

Table 2. The recent studies and proposed system for distance estimation using wireless signal.

Scheme	Objective	Data Source	Additional Equipments	LD [1]	Beaconing Type	Filtering	ML [2]	DL [3]
Research in [26]	Positioning	Wi-Fi	Wi-Fi APs	Y	Static	O	O	X
DABIL [27]	Positioning	Bluetooth	Bluetooth Beacons	Y	Static	O	O	O
Research in [28]	Positioning	Bluetooth	Bluetooth Beacons	Y	Static	X	X	X
Research in [29]	Existence Detection	Wi-Fi	Wi-Fi APs	Y	Static	X	O	X
Research in [30]	Existence Detection	Wi-Fi & Bluetooth	Wi-Fi APs & Bluetooth Beacons	Y	Static	X	O	X
Research in [31]	Existence Detection	Bluetooth	Bluetooth Beacons	N	Static	X	X	X
Research in [32]	Distance Detection	Bluetooth	Bluetooth Beacons	N	Static	O	X	X
Research in [33]	Distance Detection	Bluetooth	Bluetooth Beacons	N	Static	O	O	X
Research in [34]	Distance Detection	Bluetooth	Mobile Devices	N	Dynamic	X	O	X
Proposed System	Distance Detection	Bluetooth	Mobile Devices	N	Dynamic	O	O	O

LD [1]: Location dependency, ML [2]: Machine learning, DL [3]: Deep learning.

3. Material and Methods

In this section, we explain the proposed system with proximity detection and materials that are used to evaluate the system.

3.1. System Design

There are two requirements for providing the mobile–mobile interaction system. One is that the users should not contact an unidentified person for their safety. Therefore, users must perform interactions in a physically contactless situation. The other is that the user should find the matching person who interacts with the user on the system. To satisfy the above requirements, the system uses digital twin architecture, dynamic beaconing, location-dependent ID, and dynamic ID allocation.

3.1.1. Digital Twin Architecture

The system uses a digital twin architecture to provide secure interaction between users. The digital twin is an abstracted digital object that reflects a physical object based on information from the physical world. It is created based on the user's identity information and location information. The identity information is unique data held by each user. The location information is current location data from a user's mobile device in the physical world. This is updated by receiving location information in real time. The characteristics that digital twin maintains and updates the physical information allow the system to use the cyber-physical system (CPS) property that indicates a close correlation between physical and digital information.

3.1.2. Dynamic Beaconing

Beaconing technologies are common methods that find the user's location [35] and send the diverse information [36]. However, traditional beaconing only sends the data and their location is fixed. It leads to a decrease in the system's feasibility because the beacon must be set and deployed before the user uses it.

To increase feasibility, the system uses dynamic BLE beaconing with the mobile device. Dynamic BLE beaconing indicates that the mobile device acts as a beacon that not only transmits data but also receives it. It allows users to receive the matching person's ID as well as to send their IDs. So, not only can one user authenticate the matching person, but all users can authenticate each other. This makes it difficult for a malicious user to interfere with the authentication because the malicious user would have to interfere with the authentication process for all users, not just one user. Additionally, because BLE beacons are implemented on mobile devices, the location of the beacon is not fixed. It allows users to have and use BLE beacons easily and conveniently. Data received using dynamic BLE beaconing are passed to the user's digital twin through the mobile device.

3.1.3. Location-Dependent ID

In most digital systems, users often use IDs to prove their identity [37,38]. However, it does not guarantee that the physical person who provides the ID information is the digital user matched with the ID.

For matching the physical people and their digital twin, the system generates location-dependent IDs that represent the user's unique identity in real time, on site. The IDs are generated using the user's identity, location, and time information which are managed in the digital twin. The ID ensures the identity of the user in a certain location and time. In other words, the ID provides more detailed information about who the user is. Therefore, the users can identify the matching person by only exchanging the ID in a contactless situation.

3.1.4. Dynamic ID allocation

Dynamic ID allocation indicates that location-dependent IDs are only allocated when the user performs the authentication process. Because the IDs are generated using time and location information, the user has a unique ID each time they perform an authentication process. Of course, the time and location information are automatically managed in the user's digital twin. Therefore, if the malicious user has the user's identity information, it is difficult to interfere with the authentication process. Dynamic ID allocation using location-dependent ID improves the security of the authentication process in the system.

3.2. System Architecture

The system architecture is shown in Figure 1. The system consists of two users, the user's mobile devices and the digital twin architecture. The user's mobile device performs BLE scanning, BLE advertising, and information transmission to the digital twin from the user.

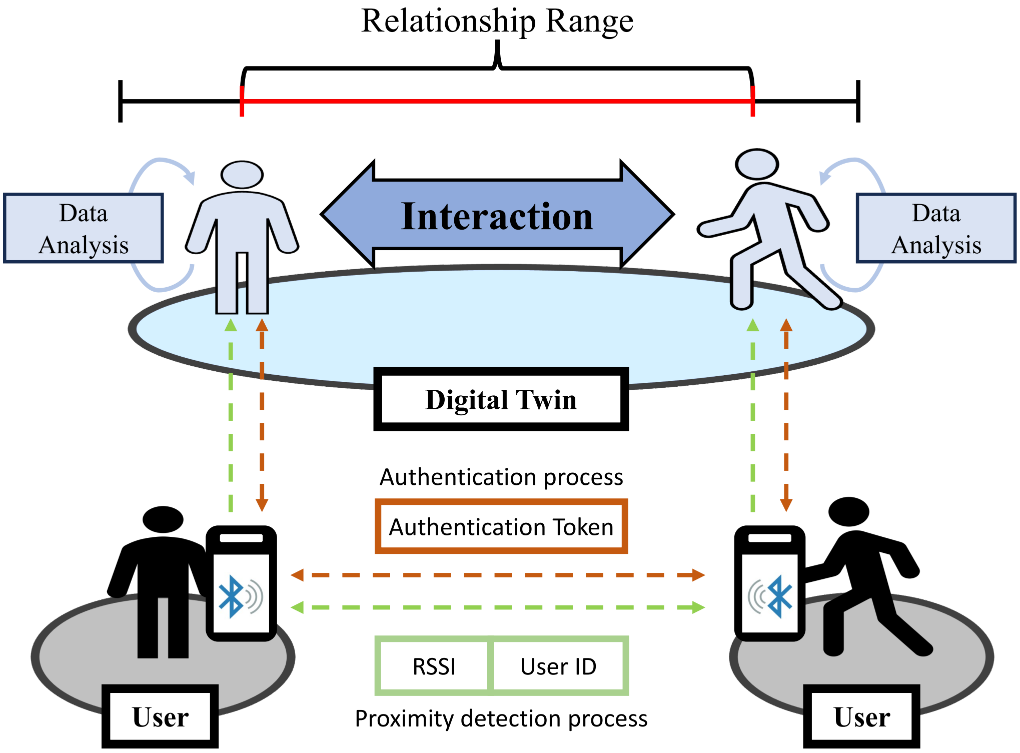


Figure 1. Mobile-based digital twin networks and inter-twin communication.

Interaction between users requires mutual recognition to find and authenticate the matching people. To recognize the user, the information representing the user's identity is exchanged and authenticated by each user. Before the users exchange their identity information, they must make sure that the matching people are within a distance where communication is possible. The system uses proximity detection using a BLE signal to measure the distance between users. The green lines in Figure 1 indicate what kinds of data are sent to each user for proximity detection. The user can distinguish who sends the BLE signal by referring to the user ID in the BLE packet. If the user checks that the sender is the matching person, the RSSI value is passed to the user's digital twin. The user's digital twin measures the proximity level with the RSSI value. Based on measuring the proximity level, it is recognized that the matching person is within the communication distance to perform the mutual authentication. After that, information to perform mutual authentication is transmitted to the digital twin. The red lines in Figure 1 show the data exchanged for the authentication process. The authentication tokens generated using user ID are sent to each user's digital twin through the physical world. The digital twins compare the received token and the token they have. If two tokens are the same, the authentication is successful.

An example case in which the system can be used is described. This is a case where the user can be identified only with the proximity level information in the interaction between users. In this case, the user knows a location where the other user is located, and the user can be confident that the person in that location is the other user. A parcel delivery service is one example. The delivery person rings the bell in front of the recipient's house. The recipient moves to the door. The delivery person and the recipient each check whether the other party is at the door with proximity detection. After that, they can perform authentication in a contactless situation.

3.3. Authentication and Proximity Detection Framework

Before the users interact with the matching users, they should carry out mutual recognition. Mutual recognition serves to find the matching users in the physical world and authenticate whether they are the matching people who interact with the user. Only after mutual recognition is complete can users safely interact with matched users. Therefore, a method to perform mutual recognition is a critical issue in the system. To explain mutual recognition, a framework is illustrated in Figure 2. This shows mutual recognition consisting of proximity detection and authentication. Proximity detection is used to measure the distance between the user and the matching person who interacts with the user. Proximity detection consists of 2 steps. The first step is a pre-processing step, which aims to reduce RSSI data fluctuations. The Kalman filter (KF), denoising autoencoder (DAE), and Kalman filter implemented autoencoder (KFAE) are used as methods that reduce fluctuation in RSSI data. Pre-processing models can be applied up to 2 times. The types of pre-processing are listed in Table 3, resulting in 1 raw and 7 pre-processed pieces of data after this step. It is worth noting that the pre-processing step is automatically carried out without additional human-based effort. This allows the system to be easily deployed to a new environment only with raw RSSI data. That is, the various pre-processed data and the classification models are automatically created and used to improve the accuracy of proximity detection. Eight classification models classify proximity levels with pre-processed data. The types of classification models are MLP, 1D-CNN, and SVM. By using 8 classification models, 8 results are obtained. The soft voting method of an ensemble technique is then applied to enhance classification accuracy. Finally, if the obtained proximity level is below the system-defined distance threshold, it means that the two users participating in the interaction are ready for mutual authentication.

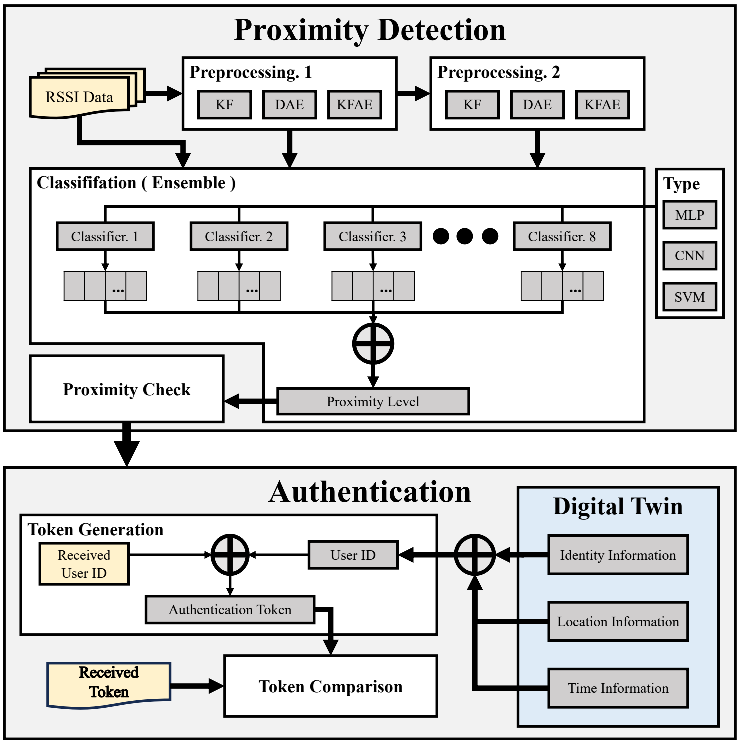


Figure 2. Framework for authentication and proximity detection process.

Table 3. Data pre-processing notation table.

Seq.	Processing 1	Processing 2	Notation
1	-	-	Raw
2	Kalman Filter	-	KF
3	Kalman Filter Using AE	-	KFAE
4	Denoising Autoencoder	-	DAE
5	Kalman Filter	Denoising Autoencoder	KF + DAE
6	Kalman Filter Using AE	Denoising Autoencoder	KFAE + DAE
7	Denoising Autoencoder	Kalman Filter	DAE + KF
8	Denoising Autoencoder	Kalman Filter Using AE	DAE + KFAE

Figure 3 shows the execution time when proximity detection is used in the system. T_{TP} is the train time of pre-processing models. T_{TC} is the train time of classification models. The point to note here is that T_{TP} and T_{TC} are the times that are executed only once before the system is applied. T_{Scan} is the time it takes to collect BLE signals to be used for proximity detection. T_{Pre} is the time taken to use pre-processing models to remove fluctuations in the collected RSSI values. T_{Cla} is the time taken to classify the 8 pre-processed data into 8 classification models. T_{Ens} is the time taken to classify the final proximity level using soft voting on the results of the 8 classification models. T_{Scan}, T_{Pre}, T_{Cla}, and T_{Ens} are spent repeatedly each time proximity detection is used.

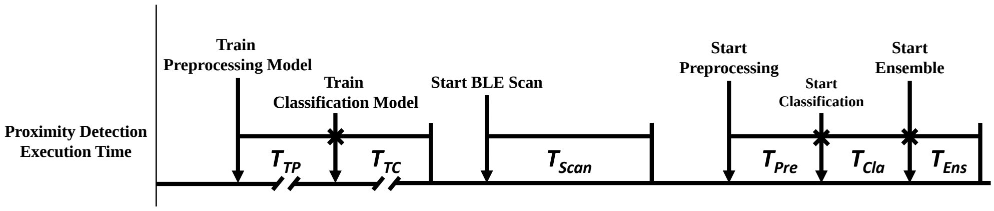


Figure 3. Time diagram about proximity detection.

After the proximity detection step is complete, a user ID is generated based on the user identity information, user location information, and time information in the digital twin. So, the ID is only valid at a specific location and time. In the authentication step, a token that is generated with the user's ID is used to authenticate the user's identity. Like this, tokens that use the ID are also only valid at a specific location and time. In the physical world, it is transmitted to the other user. When the user receives the token, it is passed to the user's digital twin. In the digital twin, the token received from the physical world is compared with the token it has. If these tokens are the same, authentication is successful.

3.4. Experiment Environment

Environments where the proximity detection experiment is conducted are described. Figure 4a shows a photo from the experiment. The Chungbuk National University building is used as an experimental environment. Cellular networks and university-provided Wi-Fi networks are also available within the building. The average temperature is 26.3 °C when we collect the data. Proximity detection in the system measures the distance between the users with mobile devices using dynamic beaconing. So, the functions of an advertiser and a scanner are performed in the mobile device with the application. The model of the mobile device is Galaxy 7 manufactured by Samsung Electronics. They have applications that advertise and scan BLE signals. Android Studio is used to make the applications. Three different environments are used to collect the BLE signals. Because the system provides mobile–mobile interaction without physical contact, proximity detection is conducted without physical contact. By placing closed doors between the mobile devices, a situation is created where there is no physical contact. Figure 4b–d show the material of doors used in the experiment. Figure 4e shows the location of the advertiser and the scanner. The distance between the advertiser and the scanner is set at 0.5 m, 1.0 m, 1.5 m, 2.0 m, 2.5 m, 3.0 m, 3.5 m, and 4.0 m. The lying mobile devices are positioned 1.0 m above the floor.

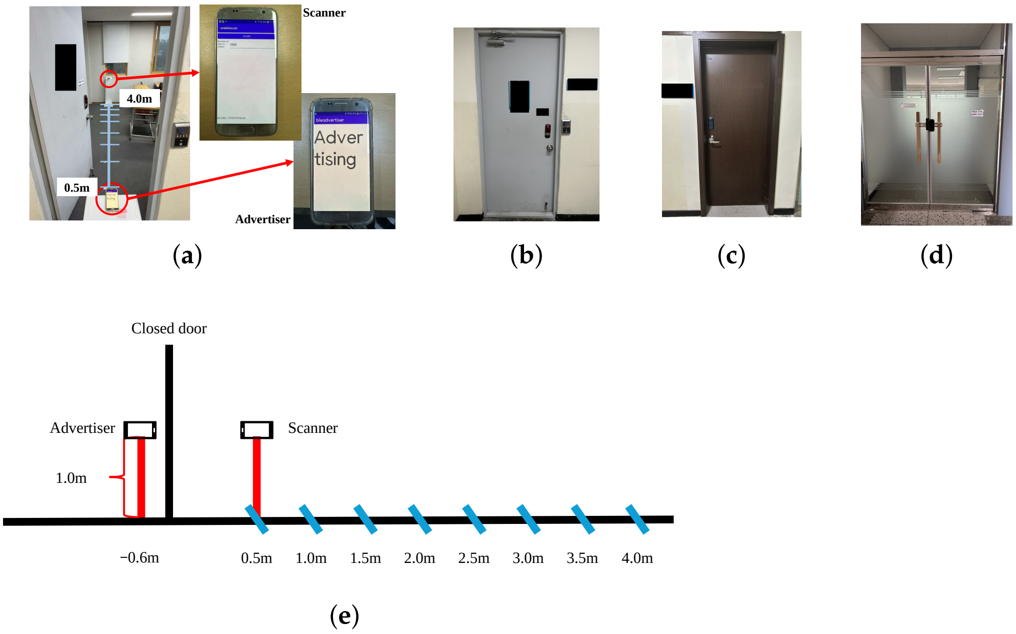


Figure 4. The experiment environments for proximity detection. (**a**) Setting of the advertiser and scanner for RSSI data collection. (**b**) Steel door. (**c**) Wood door. (**d**) Glass door. (**e**) Location of the advertiser and scanner.

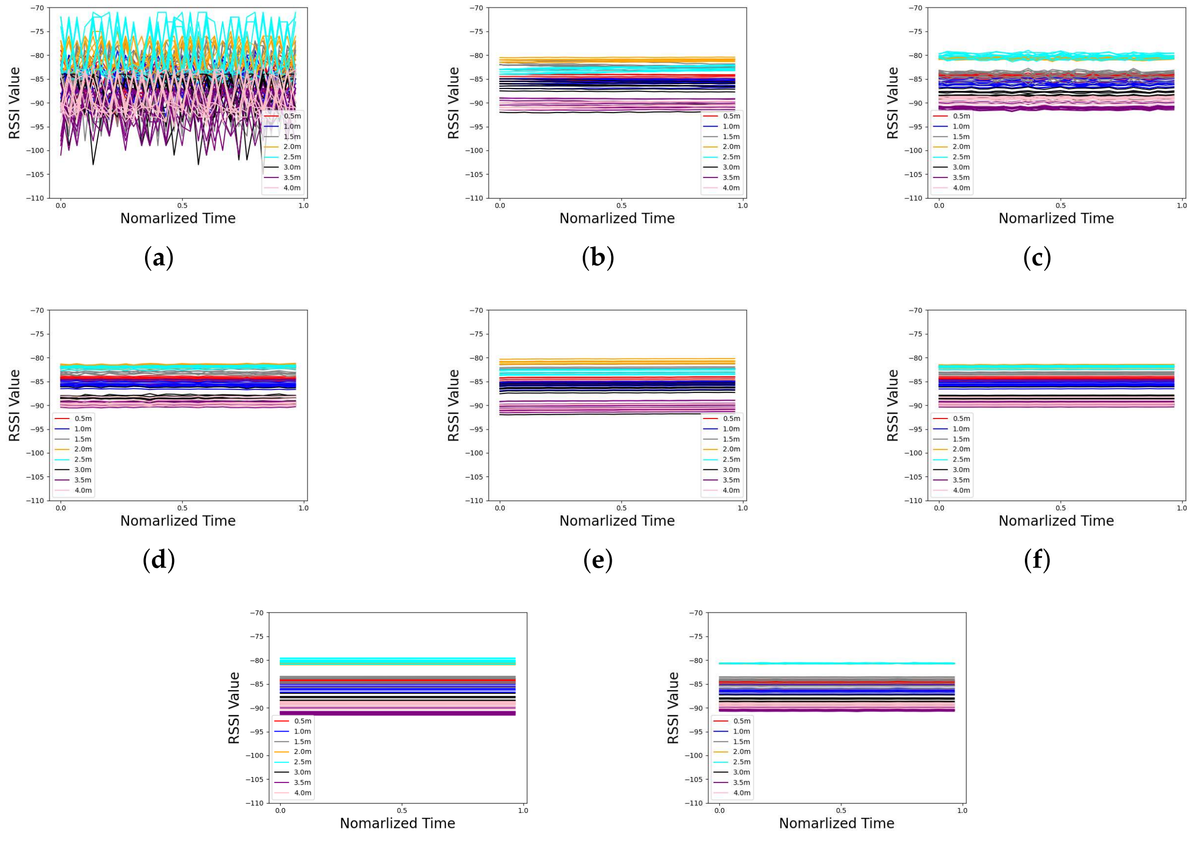
Table 4 describes the structure and parameters of the machine learning models. Ubuntu version 18.04, Python version 3.7.12, and TensorFlow library version 2.10.0 are used to implement machine learning models. In the case of SVM models, a linear kernel is used for training. Input size and output size are fixed as 30 and 8, regardless of the types of machine learning models. The 670 RSSI set is used as the train data and the 300 RSSI set is used as the test data. A total of 10 percent of the train data is used as the validation set.

Table 4. Specifications of machine learning models for training.

MLP Specification								
Layer type	Input layer	Hidden Layer	Hidden Layer	Hidden Layer	Hidden Layer	Hidden Layer	Output Layer	-
Layer size	30	64	32	16	32	64	8	-
Activation function	-	relu	relu	relu	relu	relu	softmax	-
Batch size	16							
Epoch size	150							
1D-CNN Specification								
Layer type	Input layer	Hidden Layer	Hidden Layer	Hidden Layer	Hidden Layer	Hidden Layer	Hidden Layer	Output Layer
Layer size	30	Kerenl: 3 Filter: 4	Max pooling	Kernel: 3 Filter: 2	Flatten	64	32	8
Activation function	-	relu	-	relu	-	relu	relu	softmax
Batch size	16							
Epoch size	150							

4. Research and Analysis

Before evaluating the performance of our proximity detection, we explain data processing methods to reduce fluctuation in RSSI data. RSSI data without pre-processing are called raw data. The raw data are shown in Figure 5a. As seen in Figure 5a, the fluctuation is shown in the raw data. The fluctuation can affect the accuracy of proximity classification. So, we use pre-processing models to reduce the fluctuation of RSSI data. We use three types of pre-processing: Kalman filter (KF), Kalman filter implemented using an autoencoder (KFAE), and Denoising autoencoder (DAE). These pre-processings can be applied up to two times. Notations about overall pre-processing are represented in Table 3. The results of pre-processing are represented in Figure 5. The following introduces how to implement pre-processing methods.



(**a**) (**b**) (**c**) (**d**) (**e**) (**f**) (**g**) (**h**)

Figure 5. RSSI data with various pre-processing types. (**a**) Raw RSSI data. (**b**) RSSI data with KF. (**c**) RSSI data with DAE. (**d**) RSSI data with KFAE. (**e**) RSSI data with KF + DAE. (**f**) RSSI data with KFAE + DAE. (**g**) RSSI data with DAE + KF. (**h**) RSSI data with DAE + KFAE.

4.1. Kalman Filter

The first pre-processing method is the Kalman filter. The Kalman filter is often used to reduce the fluctuation of data [39,40].

Kalman filter consists of two procedures. The first procedure is called prediction. The prediction procedure calculates the predicted action estimate and predicted estimate covariance. This procedure is represented in Equation (1).

$$\begin{aligned} \hat{x}_k^- &= A\hat{x}_{k-1} \\ P_k^- &= AP_{k-1}A^T + Q \end{aligned} \tag{1}$$

where $\hat{x}_k^-$, P_k^-, A, Q are the estimated value at time k, estimate error covariance, the action transition matrix, and noise covariance matrix, respectively.

The second procedure is called an update. The update procedure is divided into calculating Kalman gain, $\hat{x}_k$, and error covariance. These parts are shown in Equations (2)–(4).

We calculate the Kalman gain (K_k),

$$K_k = P_K^- \hat{H}(HP_k^- \hat{H} + R)^{-1} \tag{2}$$

We calculate the predicted value ($\hat{x}_k$),

$$\hat{x_k} = \hat{x_k}^- + K_k(z_k - H\hat{x_k}^-) \tag{3}$$

We calculate the predicted error covariance (P_k),

$$P_k = P_k^- - K_k HP_k^- \tag{4}$$

where K_k, P_k, x_k, z_k, H, and R are Kalman gain, predicted error covariance, predicted value, measured value, observation matrix, and noise covariance matrix, respectively.

The result of the Kalman filter is shown in Figure 5b. RSSI data in Figure 5b are obtained by applying the Kalman filter to the data in Figure 5a.

4.2. Autoencoder

Another method to reduce the fluctuation of RSSI data is a denoising autoencoder. The denoising autoencoder is one type of autoencoder. Autoencoders are often used for RSSI pre-processing [41,42]. The structure of the autoencoder consists of an encoder and a decoder. The encoder parts make input data into compressed data, which is a small dimension compared with the dimension of input data. Then, the parts of the decoder restore the original data using compressed data. This structure can be expressed as

$$X' = D(E(X)) \tag{5}$$

where X and X' are the input and output of the autoencoder, respectively. $E()$ expresses a part of the encoder process and $D()$ expresses a part of the decoder process.

The autoencoder is a type of neural network that consists of several layers. The outputs of the perceptron are calculated with the outputs of the previous layer and their weight values. Expressing this as an equation, it is

$$X^i = \phi(X^{i-1}W^{i-1} + b^i) \tag{6}$$

where X^i is the output of the ith layer. W_i and b^i are the weight matrix and the bias vector between the ith layer and the $i+1$th layer, respectively. ϕ is the activation function.

The denoising autoencoder has the same dimensions of input data and output data, but different values. The input data are data with noise and the output data are data from which noise has been removed. We define noise as fluctuation in RSSI data. Therefore, the input data and output data for denoising autoencoder learning are raw data that have fluctuation and data with fluctuations removed, respectively. Output data without fluctuations are made manually. The method to make output data is the following. Values that differ by more than three from the average value in the input data set are replaced with the average value of the input data set. A method to make output data uses Equation (7).

$$In_i = \begin{cases} In_i, & if\ |In_i - AVG(In)| \le 3 \\ AVG(In), & if\ |In_i - AVG(In)| > 3 \end{cases}, \tag{7}$$

where In_i is the ith data in the input data. $AVG(In)$ is the average value of input data.

The Kalman filter implemented autoencoder is used similarly to the denoising autoencoder. The difference with the denoising autoencoder is that data applied with the Kalman filter are output data when it is trained. The results of the denoising autoencoder and the Kalman filter implemented autoencoder are shown in Figure 5c and Figure 5d, respectively.

4.3. Ensemble

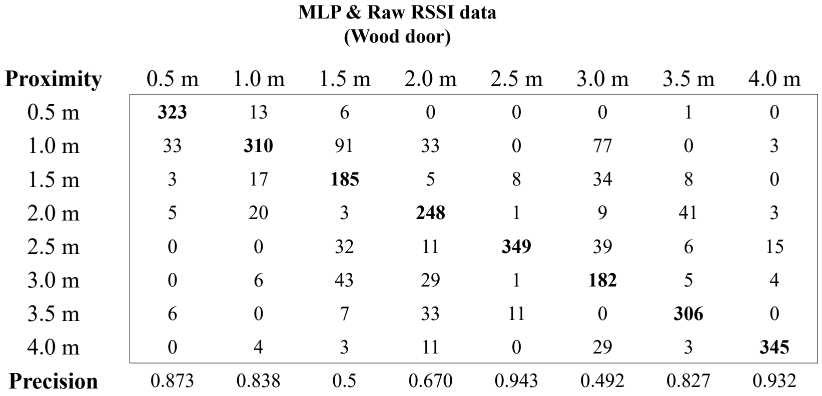
To explain the need for ensemble techniques, we show two examples of the classification results. Table 5 shows the classification accuracy of each proximity level using various pre-processing methods. The red words in the table indicate which pre-processing is applied to each proximity level to achieve the highest accuracy. In the results, we identify that the proximity level that is well classified differs depending on the type of the pre-processing method.

Table 5. Classification accuracy of each proximity level according to pre-processing type (classification model: MLP, environment: steel door).

P.P Type	0.5 m	1.0 m	1.5 m	2.0 m	2.5 m	3.0 m	3.5 m	4.0 m
Raw	**0.97**	0.85	**0.96**	0.99	0.98	**0.87**	0.96	0.81
KF	0.75	0.69	0.84	0.97	0.99	0.65	**0.98**	**0.92**
KFAE	0.72	0.63	0.34	0.97	0.32	0.59	0.91	0.39
Deno	0.59	**0.86**	0.51	0.21	0.90	0.55	0.91	0.66
KF + Deno	0.64	0.62	0.57	**1.00**	0.68	0.36	0.34	0.71
KFAE + Deno	0.67	0.74	0.20	0.87	0.47	0.47	0.94	0.19
Deno + KF	0.62	0.80	0.41	0.92	0.29	0.57	0.87	0.77
Deno + KFAE	0.57	0.82	0.42	0.00	**1.00**	0.55	0.86	0.76

The red words indicate the highest accuracy at each proximity level.

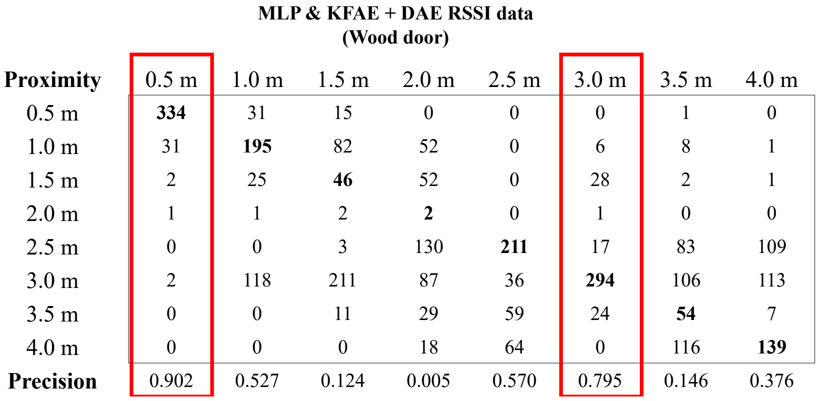
The results of proximity detection are shown in Figures 6 and 7. Figure 6 is the result of using raw data and MLP. Figure 7 results from using KFAE and DAE data and MLP. However, the results of pre-processed data have high accuracy for certain proximity levels. For example, the results in Figure 7 are more accurate than the raw data when classifying 0.5 m and 3.0 m. The proximity level that is well classified by the classifier varies depending on the pre-processing method. Therefore, improvements in accuracy can be expected by using an ensemble that combines the results of each classifier using different pre-processing data. In the ensemble process, the results of the machine learning models are aggregated into an average.



MLP & Raw RSSI data
(Wood door)

Proximity	0.5 m	1.0 m	1.5 m	2.0 m	2.5 m	3.0 m	3.5 m	4.0 m
0.5 m	**323**	13	6	0	0	0	1	0
1.0 m	33	**310**	91	33	0	77	0	3
1.5 m	3	17	**185**	5	8	34	8	0
2.0 m	5	20	3	**248**	1	9	41	3
2.5 m	0	0	32	11	**349**	39	6	15
3.0 m	0	6	43	29	1	**182**	5	4
3.5 m	6	0	7	33	11	0	**306**	0
4.0 m	0	4	3	11	0	29	3	**345**
Precision	0.873	0.838	0.5	0.670	0.943	0.492	0.827	0.932

Figure 6. Confusion matrix when using MLP and raw data from wood door.



MLP & KFAE + DAE RSSI data
(Wood door)

Proximity	0.5 m	1.0 m	1.5 m	2.0 m	2.5 m	3.0 m	3.5 m	4.0 m
0.5 m	**334**	31	15	0	0	0	1	0
1.0 m	31	**195**	82	52	0	6	8	1
1.5 m	2	25	**46**	52	0	28	2	1
2.0 m	1	1	2	**2**	0	1	0	0
2.5 m	0	0	3	130	**211**	17	83	109
3.0 m	2	118	211	87	36	**294**	106	113
3.5 m	0	0	11	29	59	24	**54**	7
4.0 m	0	0	0	18	64	0	116	**139**
Precision	0.902	0.527	0.124	0.005	0.570	0.795	0.146	0.376

Figure 7. Confusion matrix when using MLP and KFAE + DAE data from wood door. The red rectangles indicate that the accuracy using pre-processing has improved compared to the accuracy using raw data at specific proximity levels.

5. Results

In this section, we show the performance of the proposed proximity detection scheme that uses pre-processing, machine learning models, and ensemble techniques. We compare the proposed proximity detection without pre-processing and ensemble techniques to describe performance improvement. The differences in accuracy and execution time are shown in the section.

5.1. Accuracy

Figure 8 represents results of accuracy using raw data with machine learning models and the log distance path loss model (LDPL) [43,44]. The LDPL is a common method to measure distance from the signal transmitters with their RSSI value. As a result, machine learning models have better accuracy than the LDPL model. This is because LDPL is based on the assumption that RSSI values decrease with increasing distance. Because the sources that interfere with the signal propagation make sure that the RSSI value may not decrease as the distance increases, a decrease in the accuracy of the LDPL model occurs.

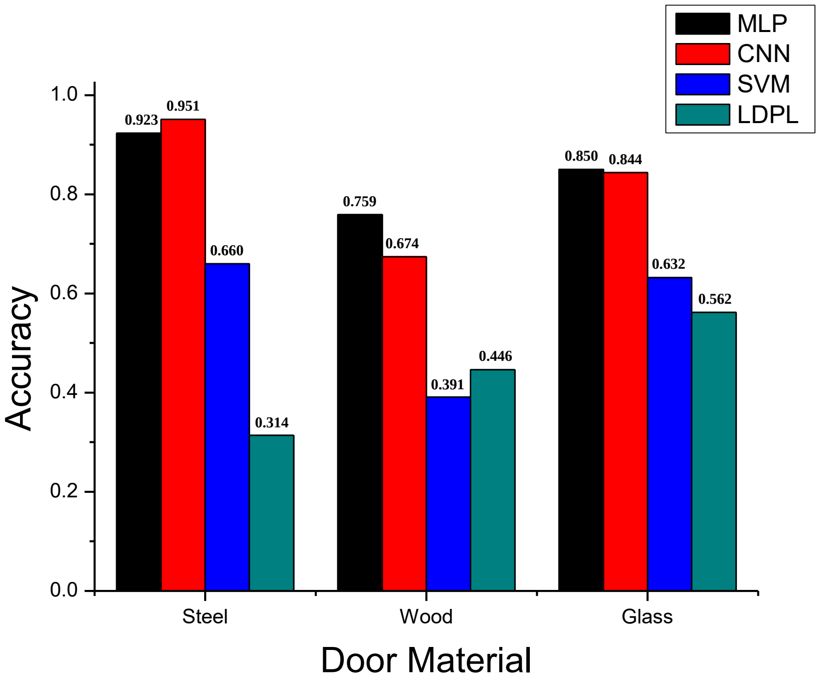


Figure 8. Accuracy for each classification type when using raw data according to door material.

Deep learning models such as MLP and 1D-CNN generally have higher accuracy than SVM. Additionally, it is confirmed that the accuracy varies depending on the material of the door. Next, we show the accuracy of proximity detection using the proposed pre-processing and ensemble.

Figures 9a and 10a show proximity accuracy using the ensemble and the error reduction rate compared with using raw data when the steel door is between the scanner and the advertiser. As a result, MLP and 1D-CNN have better classification performance than SVM. Also, ensembles can improve accuracy. It was seen that the ensemble can improve accuracy regardless of the type of classifier. Likewise, Figures 9b and 10b represent the accuracy and the error reduction rate when the wood door is between the scanner and the advertiser. Also, MLP and 1D-CNN show better classification performance than SVM. In the case of MLP and 1D-CNN, the ensemble can improve accuracy. However, in the case of SVM, the accuracy was seen to decrease. The glass door between the scanner and the advertiser produces similar results as the wood door. Figures 9c and 10c represent the accuracy and the error reduction rate when the wood door is between the scanner and the advertiser. In this case, MLP and 1D-CNN show higher accuracy than SVM, and it was confirmed that accuracy improved when the ensemble was used. However, the accuracy of SVM is decreased using the ensemble.

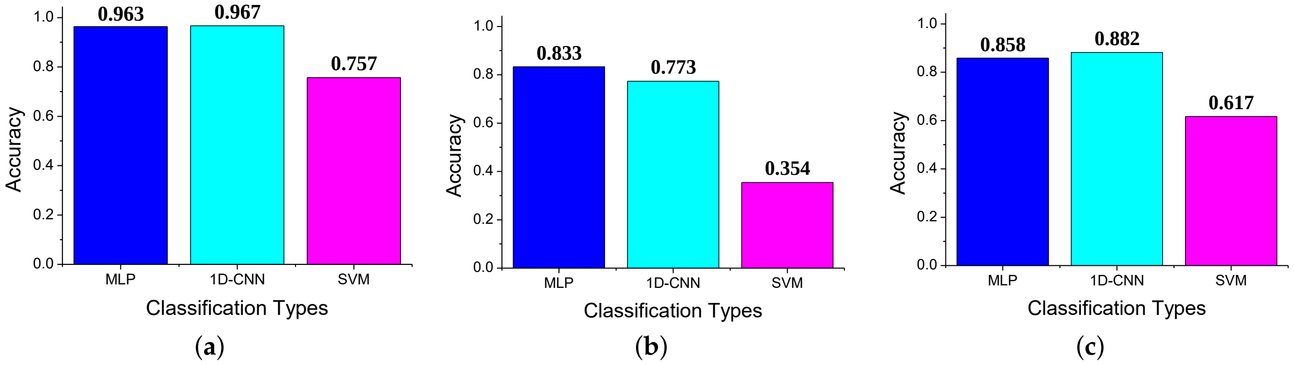


Figure 9. Proximity detection accuracy applied by ensemble for each classification model types. (**a**) Steel door. (**b**) Wood door. (**c**) Glass door.

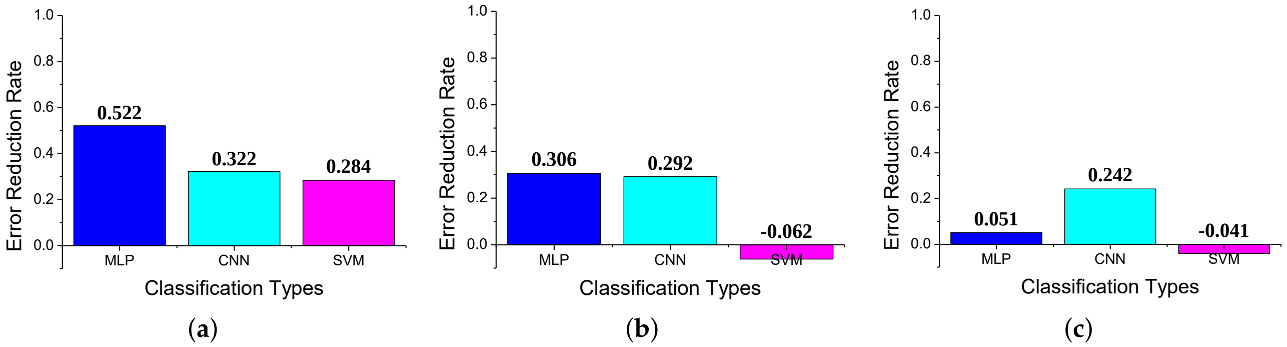


Figure 10. Error reduction rate of ensemble compared with the model using raw data. (**a**) Steel door. (**b**) Wood door. (**c**) Glass door.

As an aspect of accuracy, deep learning models have better performance than machine learning models, regardless of the environment. Additionally, it has been confirmed that ensemble provides good effects in deep learning models such as MLP and 1D-CNN. Since the ensemble process uses the results of eight classifiers, the performance of the classifiers is also important. Although the accuracy of a classifier using pre-processed data is lower than that of a classifier using raw data, we confirmed in the previous section that the accuracy of classifying a specific distance can be high.

5.2. Execution Time

The execution time of proximity detection is affected by the classification model type and ensemble. We compare the execution time that varies depending on the experimental environment, whether an ensemble is used, and each classification model.

The results of the execution time for each experimental environment are shown in Tables 6–8. Since the ensemble applies various pre-processing to raw data, T_{TP} and T_{Pre} data exist. In addition, T_{Ens} data are the ensemble process time using the results of each single classification model. Therefore, the fact that the ensemble takes longer to run than the raw model is determined.

Table 6. Execution time for proximity detection when the environment is a steel door.

Model Type	T_{TP}	T_{TC}	T_{Scan}	T_{Pre}	T_{Cla}	T_{Ens}
Raw_MLP	0 ms	16,511 ms	4000 ms	0 ms	38 ms	0 ms
Ensemble_MLP	21,337 ms	16,925 ms	4000 ms	83 ms	40 ms	0.29 ms
Raw_1D-CNN	0 ms	18,507 ms	4000 ms	0 ms	40 ms	0 ms
Ensemble_1D-CNN	21,337 ms	18,832 ms	4000 ms	83 ms	40 ms	0.29 ms
Raw_SVM	0 ms	356 ms	4000 ms	0 ms	0.2 ms	0 ms
Ensemble_SVM	21,337 ns	427 ms	4000 ms	83 ms	1.3 ms	0.06 ms

Table 7. Execution time for proximity detection when the environment is a wood door.

Model Type	T_{TP}	T_{TC}	T_{Scan}	T_{Pre}	T_{Cla}	T_{Ens}
Raw_MLP	0 ms	16,497 ms	4000 ms	0 ms	38 ms	0 ms
Ensemble_MLP	20,891 ms	16,898 ms	4000 ms	82 ms	40 ms	0.29 ms
Raw_1D-CNN	0 ms	18,425 ms	4000 ms	0 ms	39 ms	0 ms
Ensemble_1D-CNN	20,891 ms	18,749 ms	4000 ms	82 ms	39 ms	0.29 ms
Raw_SVM	0 ms	657 ms	4000 ms	0 ms	0.21 ms	0 ms
Ensemble_SVM	20,891 ms	751 ms	4000 ms	82 ms	1.2 ms	0.06 ms

Table 8. Execution time for proximity detection when the environment is a glass door.

Model Type	T_{TP}	T_{TC}	T_{Scan}	T_{Pre}	T_{Cla}	T_{Ens}
Raw_MLP	0 ms	16,579 ms	4000 ms	0 ms	38 ms	0 ms
Ensemble_MLP	21,357 ms	17,026 ms	4000 ms	83 ms	40 ms	0.29 ms
Raw_1D-CNN	0 ms	18,568 ms	4000 ms	0 ms	39 ms	0 ms
Ensemble_1D-CNN	21,357 ms	18,929 ms	4000 ms	83 ms	40 ms	0.29 ms
Raw_SVM	0 ms	459 ms	4000 ms	0 ms	0.19 ms	0 ms
Ensemble_SVM	21,357 ms	522 ms	4000 ms	83 ms	1 ms	0.06 ms

As a result, T_{Cls} and T_{Ens} of SVM take a shorter time than MLP and 1D-CNN. In addition, methods using ensembles take longer than methods using raw data because they use pre-processing. Considering the time results and accuracy results, it can be seen that there is a trade-off in accuracy and time between deep learning models such as MLP and 1D-CNN and machine learning models such as SVM. Methods using raw data and methods using ensembles also have the above trade-off.

6. Discussion

Because the system determines whether an authentication process starts based on measured proximity levels, the performance improvement of proximity detection increases the system's stability. Results of proximity detection experiments show that using an ensemble and various pre-processing results in reduced error rates compared to only using raw data. We find that MLP and 1D-CNN are better than SVM in the system with pre-processing and ensemble. Additionally, execution times for using raw data and using the ensemble are shown. So, a trade-off between accuracy and time is found.

Although the stability of the system is enhanced by increasing the accuracy of the proximity detection, potential challenges remain. Proximity detection may decrease the accuracy when the user moves while the mobile device scans the BLE signal. Because human movement affects the RSSI value, the proximity level measured using RSSI may be measured incorrectly. To reduce this negative effect on the user's movement, the system recommends that the authentication process should be carried out after the user stops at the location where the user promised to meet the matching person. Also, proximity detection can be used to determine that the user is within a specific distance. For example, let us assume the system requires that a matched person be within 2.0 m of the user. Then, 1.0 m is the same as 2.0 m and 2.5 m is the same as 3.0 m. This can make the system become error-tolerant.

The system can be used for a traditional mobile–mobile interaction system that supports a user-matching system. In the case of the delivery service, a delivery person matches with a buyer. When the delivery person arrives at the buyer's house, the delivery person and the buyer use the system for mutual recognition in a contactless situation. In the case of the ride-hailing system, a driver matches with a passenger. After the driver arrives at the location where the passenger requested, the passenger finds the car of the driver. The passenger finds the car of the driver and confirms that it is the car they called by using mutual recognition of the system. By using the system, users can recognize each other in a contactless situation. This reduces risks, such as crimes, infectious diseases, etc., that arise in a contact situation. Consequently, the safety of the user is increased with the system.

7. Conclusions

The paper proposes a novel contactless interaction system with digital twin architecture. To provide contactless interaction, proximity detection is used to find the matching people in the physical world. Based on dynamic beaconing, we improve the system's feasibility. Additionally, location-dependent ID and dynamic ID allocation improve the security of the system by generating a unique ID using a user's identity, location, and time information managed by a digital twin. To summarize, the system allows users to interact with each other without risks of contact situations.

In the experiment, we find that the type of pre-processing affects the accuracy of proximity detection. In other words, the accuracy of specific proximity levels, which are measured by the models, increases according to the type of pre-processing. Based on this fact, we use the ensemble that aggregates the results of models to improve the performance of proximity detection. Future work involves using various pre-processing methods and comparing them to maximize these effects.

Author Contributions: Conceptualization, W.P., H.A., Y.Y. and S.P.; methodology, W.P., H.A., Y.Y. and S.P.; software, W.P.; validation, W.P., H.A., Y.Y. and S.P.; formal analysis, W.P., H.A., Y.Y. and S.P.; investigation, W.P., H.A., Y.Y. and S.P.; resources, W.P.; data curation, W.P.; writing—original draft preparation, W.P.; writing—review and editing, H.A., Y.Y. and S.P.; visualization, W.P., H.A., Y.Y. and S.P.; supervision, Y.Y. and S.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kim, S.; Park, S.; Lee, S.-H.; Yang, T. Smart Parking with Learning-aided User Activity Sensing Based on Edge Computing. In Proceedings of the 2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 9–12 January 2021; pp. 1–2.
2. Lee, C.; Park, S.; Yang, T.; Lee, S.-H. Smart Parking with Fine-Grained Localization and User Status Sensing Based on Edge Computing. In Proceedings of the 2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall), Honolulu, HI, USA, 22–25 September 2019; pp. 1–5.
3. Árvai, L. Mobile phone based indoor navigation system for blind and visually impaired people: VUK—Visionless supporting framework. In Proceedings of the 2018 19th International Carpathian Control Conference (ICCC), Szilvasvarad, Hungary, 28–31 May 2018; pp. 383–388.
4. Jain, M.; Rahul, R.C.P.; Tolety, S. A study on Indoor navigation techniques using smartphones. In Proceedings of the 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Mysore, India, 22–25 August 2013; pp. 1113–1118.
5. Al-Sadi, A.; Al-Theiabat, H.; Awad, F. Smartphone-assisted location identification algorithm for search and rescue services. In Proceedings of the 2017 8th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, 4–6 April 2017; pp. 276–281.
6. Ma, X.; Huang, Q.-Y.; Shu, X.-M. A New Localization Algorithm of Mobile Phone for Outdoor Emergency Rescue. In Proceedings of the 2013 Third International Conference on Intelligent System Design and Engineering Applications, Hong Kong, China, 16–18 January 2013; pp. 124–127.
7. Amies, N. Fake Delivery Men Rob Woman in Her Home in Schaerbeek. Available online: https://www.thebulletin.be/fake-delivery-men-rob-woman-her-home-schaerbeek (accessed on 30 July 2023).
8. Adreani, L.; Bellini, P.; Fanfani, M.; Nesi, P.; Pantaleo, G. Smart City Digital Twin Framework for Real-Time Multi-Data Integration and Wide Public Distribution. *IEEE Access* **2024**, *12*, 76277–76303. [CrossRef]
9. Liu, J.; Wang, Y.; Gong, W.; Liu, H.; Xu, Y.; Kou, H. A Smart Data-Driven Multi-Level Synchronous Digital Twin Model for Vehicle-Assisted Driving. *IEEE Trans. Consum. Electron.* **2024**, *70*, 4037–4049. [CrossRef]
10. Zhao, C.; Xu, X.; Zhang, D.; Wei, Y. Intelligent Human-Machine Interaction Based on Digital Twin and Virtual Reality. In Proceedings of the 2023 5th International Conference on Robotics, Intelligent Control and Artificial Intelligence (RICAI), Hangzhou, China, 1–3 December 2023; pp. 374–378.
11. Johnson, Z.; Saikia, M.J. Digital Twins for Healthcare Using Wearables. *Bioengineering* **2024**, *11*, 606. [CrossRef] [PubMed]
12. Rakhra, M.; Singh, A.; Singh, D.; Shruti. Digital Signature Verification In Cloud Computing. In Proceedings of the 2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 14–15 March 2024; pp. 1–6.
13. Rai, A.K.; Singh, M.; Sudheendramouli, H.C.; Panwar, V.; Balaji, N.A.; Kukreti, R. Digital Signature for Content Authentication. In Proceedings of the 2023 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI), Chennai, India, 25–26 May 2023; pp. 1–6.
14. Gupta, B.B.; Gaurav, A.; Hsu, C.-H.; Jiao, B. Identity-Based Authentication Mechanism for Secure Information Sharing in the Maritime Transport System. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 2422–2430. [CrossRef]
15. Zhang, Y.; Sun, L.; Liu, X.; Ai, Y.; Zhao, Y. Terminal ID Authentication System Based on Digital Certificate. In Proceedings of the 2023 International Conference on Data Science & Informatics (ICDSI), Bhubaneswar, India, 12–13 August 2023; pp. 111–115.
16. Mukhandi, M.; Damião, F.; Granjal, J.; Vilela, J.P. Blockchain-based Device Identity Management with Consensus Authentication for IoT Devices. In Proceedings of the 2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 8–11 January 2022; pp. 433–436.
17. Liu, L.; Omote, K. Efficient Authentication System Based On Blockchain Using eID card. In Proceedings of the 2023 IEEE International Conference on Blockchain (Blockchain), Danzhou, China, 17–21 December 2023; pp. 166–171.
18. Amazon. Amazon Key. Available online: https://www.amazon.com/Amazon-Key-In-Garage-Delivery/b?ie=UTF8&node=21222091011 (accessed on 20 August 2023).
19. Li, H.; Niu, Y.; Yi, J.; Li, H. Securing Offline Delivery Services by Using Kerberos Authentication. *IEEE Access* **2018**, *6*, 40735–40746. [CrossRef]
20. AlQahtani, A.A.S.; Alshayeb, T.; Nabil, M.; Patooghy, A. Leveraging Machine Learning for Wi-Fi-Based Environmental Continuous Two-Factor Authentication. *IEEE Access* **2024**, *12* 13277–13289. [CrossRef]
21. Nimura, M.; Kanai, K.; Katto, J. Accuracy evaluations of real-time LiDAR-based indoor localization system. In Proceedings of the 2023 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 6–8 January 2023; pp. 1–5.
22. Wang, L.; Liu, Z.; Li, H. Robust and Real-Time Outdoor Localization Only with a Single 2-D LiDAR. *IEEE Sens. J.* **2022**, *22*, 24516–24525. [CrossRef]
23. Chen, W.; Xu, J.; Zhao, X.; Liu, Y.; Yang, J. Separated Sonar Localization System for Indoor Robot Navigation. *IEEE Trans. Ind. Electron.* **2021**, *68*, 6042–6052. [CrossRef]
24. Jia, T.; Shen, X.; Wang, H. Multistatic Sonar Localization With a Transmitter. *IEEE Access* **2019**, *7*, 111192–111203. [CrossRef]

25. Kang, D.; Kum, D. Camera and Radar Sensor Fusion for Robust Vehicle Localization via Vehicle Part Localization. *IEEE Access* **2020**, *8*, 75223–75236. [CrossRef]
26. Singh, V.; Aggarwal, G.; Ujwal, B.V.S. Ensemble based real-time indoor localization using stray WiFi signal. In Proceedings of the 2018 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 12–14 January 2018; pp. 1–5.
27. Xiao, C.; Yang, D.; Chen, Z.; Tan, G. 3-D BLE Indoor Localization Based on Denoising Autoencoder. *IEEE Access* **2017**, *5* 12751–12760. [CrossRef]
28. Faragher, R.; Harle, R. Location Fingerprinting with Bluetooth Low Energy Beacons. *IEEE J. Sel. Areas Commun.* **2015**, *33*, 2418–2428. [CrossRef]
29. Van Hyfte, Z.; Zakhor, A. Immediate Proximity Detection Using Wi-Fi–Enabled Smartphones. In Proceedings of the 2021 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Lloret de Mar, Spain, 29 November–2 December 2021; pp. 1–8.
30. Kalabakov, S.; Švigelj, A.; Javornik, T. Smartphone Proximity Detection Using WiFi and BLE Fingerprinting. In Proceedings of the 2022 International Balkan Conference on Communications and Networking (BalkanCom), Sarajevo, Bosnia and Herzegovina, 22–24 August 2022; pp. 36–40.
31. Ng, P.C.; She, J.; Park, S. High Resolution Beacon-Based Proximity Detection for Dense Deployment. *IEEE Trans. Mob. Comput.* **2018**, *17*, 1369–1382. [CrossRef]
32. Mackey, A.; Spachos, P.; Song, L.; Plataniotis, K.N. Improving BLE Beacon Proximity Estimation Accuracy through Bayesian Filtering. *IEEE Internet Things J.* **2020**, *7*, 3160–3169. [CrossRef]
33. Lam, C.H.; Jeon, K.E.; Wong, J.; She, S. Distance Estimation Using BLE Beacon on Stationary and Mobile Objects. *IEEE Internet Things J.* **2022**, *9*, 4928–4939. [CrossRef]
34. Su, Z.; Pahlavan, K.; Agu, E. Performance Evaluation of COVID-19 Proximity Detection Using Bluetooth LE Signal. *IEEE Access* **2021**, *9*, 38891–38906. [CrossRef] [PubMed]
35. Spachos, P.; Plataniotis, K.N. BLE Beacons for Indoor Positioning at an Interactive IoT-Based Smart Museum. *IEEE Syst. J.* **2020**, *14*, 3483–3493. [CrossRef]
36. Rao, A.S.; Sharma, A.V.; Narayan, C.S. A context aware system for an IoT-based smart museum. In Proceedings of the 2017 2nd International Multidisciplinary Conference on Computer and Energy Science (SpliTech), Split, Croatia, 12–14 July 2017; pp. 1–5.
37. Tsai, T.-T.; Chuang, Y.-H.; Tseng, Y.-M.; Huang, S.-S.; Hung, Y.-H. A Leakage-Resilient ID-Based Authenticated Key Exchange Protocol with a Revocation Mechanism. *IEEE Access* **2021**, *9*, 128633–128647. [CrossRef]
38. Hakim, L.; Kristanto, S.P.; Yusuf, D.; Fanani, N.Z. Empowering Library Attendance System with IoT-Enabled Student ID Cards. In Proceedings of the 2023 International Conference on Advanced Mechatronics, Intelligent Manufacture and Industrial Automation (ICAMIMIA), Surabaya, Indonesia, 14–15 November 2023; pp. 673–678.
39. Nazemi, M.S.; Hakimnejad, H.; Azimifar, Z. PCG denoising using AR-based Kalman Filter. In Proceedings of the 2021 29th Iranian Conference on Electrical Engineering (ICEE), Tehran, Iran, 18–20 May 2021; pp. 902–906.
40. Xie, Y.; Jiang, L. RSSI Indoor Positioning Algorithm Based on Kalman Filtering. In Proceedings of the 2024 5th International Conference on Electronic Communication and Artificial Intelligence (ICECAI), Shenzhen, China, 31 May–2 June 2024; pp. 139–142.
41. Ju, C.; Yoo, J. Machine Learning for Indoor Localization Without Ground-truth Locations. In Proceedings of the 2023 13th International Conference on Indoor Positioning and Indoor Navigation (IPIN), Nuremberg, Germany, 25–28 September 2023; pp. 1–5.
42. Cao, W.; Huang, J.; Zeng, M. RSSI-Based Trajectory Prediction for Intelligent Indoor Localization. In Proceedings of the 2023 IEEE 23rd International Conference on Communication Technology (ICCT), Wuxi, China, 20–22 October 2023; pp. 445–450.
43. Huan, H.; Wang, K.; Xie, Y.; Zhou, L. Indoor Location Fingerprinting Algorithm Based on Path Loss Parameter Estimation and Bayesian Inference. *IEEE Sens. J.* **2023**, *23*, 2507–2521. [CrossRef]
44. Moradbeikie, A.; Azevedo, R.; Jesus, C.; Lopes, S.I. RSSI-Based Localization in Industrial Environments: A Wi-Fi/BLE Hybrid Approach. In Proceedings of the 2024 IEEE International Conference on Industrial Technology (ICIT), Bristol, UK, 25–27 March 2024; pp. 1–6.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.