

Article

A Novel Valued Tolerance Rough Set and Decision Rules Method for Indoor Positioning Using WiFi Fingerprinting

Ninh Duong-Bao ^{1,2}, Jing He ¹, Luong Nguyen Thi ³, Khanh Nguyen-Huu ^{4,*} and Seon-Woo Lee ⁵

¹ College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China; duongbaoninh@hnu.edu.cn (N.D.-B.); jhe@hnu.edu.cn (J.H.)

² Faculty of Mathematics and Informatics, Dalat University, Dalat 66100, Vietnam

³ Faculty of Information Technology, Dalat University, Dalat 66100, Vietnam; luongnt@dlu.edu.vn

⁴ Department of Electronics and Telecommunications, Dalat University, Dalat 66100, Vietnam

⁵ Division of Software, Hallym University, Chuncheon 24252, Korea; senu@hallym.ac.kr

* Correspondence: kxanhnh@dlu.edu.vn

Abstract: In recent years, due to the ubiquitous presence of WiFi access points in buildings, the WiFi fingerprinting method has become one of the most promising approaches for indoor positioning applications. However, the performance of this method is vulnerable to changes in indoor environments. To tackle this challenge, in this paper, we propose a novel WiFi fingerprinting method that uses the valued tolerance rough set theory-based classification method. In the offline phase, the conventional received signal strength (RSS) fingerprinting database is converted into a decision table. Then a new fingerprinting database with decision rules is constructed based on the decision table, which includes the credibility degrees and the support object set values for all decision rules. In the online phase, various classification levels are applied to find out the best match between the RSS values in the decision rules database and the measured RSS values at the unknown position. The experimental results compared the performance of the proposed method with those of the nearest-neighbor-based and the random statistical methods in two different test cases. The results show that the proposed method greatly outperforms the others in both cases, where it achieves high accuracy with 98.05% of right position classification, which is approximately 50.49% more accurate than the others. The mean positioning errors at wrong estimated positions for the two test cases are 1.71 m and 1.99 m, using the proposed method.

Keywords: indoor positioning; WiFi fingerprinting; RSS; rough set; valued tolerance; decision rules



Citation: Duong-Bao, N.; He, J.; Thi, L.N.; Nguyen-Huu, K.; Lee, S.-W. A Novel Valued Tolerance Rough Set and Decision Rules Method for Indoor Positioning Using WiFi Fingerprinting. *Sensors* **2022**, *22*, 5709. <https://doi.org/10.3390/s22155709>

Academic Editor: Alejandro Santos Martínez Sala

Received: 22 June 2022

Accepted: 27 July 2022

Published: 30 July 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Over the past ten years, location-based services have rapidly developed. In these services, positioning is one of the most indispensable approaches. For outdoor environments, the global navigation satellite system (GNSS) can provide very reasonable positioning results, with an average error of 3–5 m [1]. However, this system is limited outdoors since it cannot guarantee the same accuracy indoors, where the satellite signals are reflected or blocked by the walls of buildings. Thus, there is an extreme demand to develop indoor positioning systems (IPs).

Currently, there are various technologies used to track the position of a mobile user indoors. These technologies can be classified into two groups: building independent and building dependent. The first group includes the IPs that do not rely on the infrastructure of a building to determine the user's position. Some examples for this group are dead reckoning [2] and image-based [3] technologies. On the other hand, the IPs that relate to the building where they are operated belong to the second group. For this group, we can divide it into two subgroups: the dedicated infrastructure required and the building's infrastructure utilized. The former subgroup is defined as the IPs that need the unpopular

infrastructure to be set up in a building. Radio frequency identification [4], acoustic signal [5], or visible light communication [6] are some examples belonging to this group. The latter subgroup utilizes the common infrastructure in the building to create the IPSs such as Bluetooth [7,8] and WiFi [9,10]. Among these technologies, WiFi-based positioning [11–13] has drawn tremendous attention due to the widespread of WiFi infrastructure in indoor environments such as office buildings, hospitals, malls, etc. Moreover, the high popularization of smart devices such as smartphones or tablets and the compatible implementation of these devices with WiFi systems also leads to the large deployment of this technology for IPS. To enhance the performance of the WiFi-based positioning system, there are hybrid systems that combine the result of the WiFi with other technologies, such as pedestrian dead reckoning [2,9], camera [14], or magnetic field [15].

There are two main approaches for WiFi-based positioning: geometric and fingerprinting [16]. For the former approach, some common conventional positioning methods are the time of arrival [17], time difference of arrival [18], angle of arrival [19], etc. The positioning accuracy of these methods depends on the line-of-sight condition, which is hard to keep indoors, where there are many physical obstacles, such as walls or doors. For the latter one, the WiFi fingerprinting method is another popular method due to its low cost and ease of implementation on smart devices. However, the reflections, the multipath interference, or the changes in the environmental conditions could greatly degrade the performance of this method. Therefore, achieving a reliable and highly accurate WiFi fingerprinting-based positioning method is a challenging problem.

The traditional WiFi fingerprinting method has two phases: the offline and the online phases. In the offline phase, the radio map is built by collecting the RSS values from the available APs at different reference points (RPs) in a target area. The unique set of RSS values at one RP is defined as a fingerprint of that position. In the online phase, by comparing the measured RSS values at one unknown position with the RSS values from the radio map using different matching methods, the user's position can be estimated.

The conventional rough set theory was first proposed by Pawlak [20] as an extension of classical set theory to deal with vague information. This theory conceives that information and knowledge in information systems are expressed via indiscernibility relations (equivalence relations) between objects in the object set. The rough set theory is used for big data mining and knowledge reduction [21,22] and can be applied in various fields such as information systems [23] or real estate market analysis [24,25]. To give a more flexible way to the rough set theory to handle the indiscernibility relation, Stefanowski and Tsoukias [26] introduced the valued tolerance relation and proposed the valued tolerance rough set and decision rules method (VTRS–DR). This method is often applied as a support tool in decision-making systems. Rough set theory based on valued tolerance relation calculates the valued tolerance relation for all objects in the decision table, resulting in a valued tolerance relation matrix where the values belong to $[0, 1]$. This is the extension of the conventional rough set theory proposed by Pawlak.

When applied to WiFi fingerprinting-based indoor positioning, the VTRS–DR works not only in the offline phase but also in the online phase. In the offline phase, the conventional RSS fingerprinting database is converted into a decision table where the RSS values are classified into different decision classes which correspond to the predefined RPs. Each tolerance granule corresponding to each object (i.e., one row in the decision table) is calculated based on the valued tolerance relation matrix. After calculating the values of the membership degrees of objects belonging to lower (upper) approximations of each decision class, the credibility degrees of decision rules (i.e., objects in the decision table) are also calculated. Besides that, the support object set values of decision rules are defined by the tolerance granule of objects. A new fingerprinting database with decision rules is constructed from the decision table that includes the credibility degrees and the support object set values for all decision rules. In the online phase, the valued tolerance relation is calculated between the RSS values in the new database and the measured RSS values at an unknown position. The best decision class (i.e., one RP) is chosen among a set of RPs

via the proposed method by comparing various components such as the valued tolerance relations, the credibility degrees, the support object set values, and the Euclidean distances; thus, the user's position can be estimated. In our work, it means estimating the unknown position as the RP closest to that position.

To deal with the instability in RSS values from APs, in this paper, a novel VTRS-DR-based WiFi fingerprinting for IPS is proposed. The VTRS-DR method provides an efficient computational structure and helps to solve the indoor positioning problem with high accuracy. The main contributions are figured out as follows:

- The VTRS-DR method is applied in the offline phase of the WiFi fingerprinting method by creating a new fingerprinting database with decision rules that includes the credibility degrees and the support object set values for all rules.
- The VTRS-DR method is also applied in the online phase of the WiFi fingerprinting method by estimating the user's position based on the fingerprinting decision rules database and the measured RSS value to determine the user's position.
- For performance evaluation, the proposed method is compared with the nearest neighbor-based and the random statistical methods to prove its superior positioning accuracy and robustness.

The rest of this paper is organized as follows. Section 2 discusses the related works. Section 3 describes the basis of the VTRS-DR method. Section 4 introduces the proposed positioning method. Experimental results are displayed in Section 5, and the conclusion is given in Section 6.

2. Related Works

Generally, there are two categories of WiFi fingerprint-based matching algorithms: deterministic and probabilistic. The deterministic algorithm is very common due to its ease of implementation and its possibility to work well in real time. The nearest neighbor (NN), KNN, and weighted KNN (WKNN) [27,28] are some of the popular deterministic algorithms. To handle the fluctuation of WiFi RSS values, as well as to normalize them, Ninh et al. [10] introduced the random statistical method in the offline phase. The method helped to create a standardized fingerprinting database. In the online phase, to figure out the unknown position of a user, the authors applied the Mahalanobis distance as the matching algorithm to improve the positioning accuracy. The experiments were conducted in different setup conditions in an office room. As a result, the maximum positioning error was only 0.75 m. In Reference [29], instead of using the common Euclidean distance in NN-based algorithms, Duong-Bao et al. implemented five distance measures and compared the positioning results in different settings such as changing the number of setup APs or changing the grid spacing between two RPs. The experimental results showed that the Chi-Squared distance is the best measure with a mean error of 1.17 m in two test cases. Even though these deterministic algorithms could reduce computational complexity, they used only a single reference fingerprint for each RP (i.e., a set of mean RSS values from available AP) for finding the best match between the measured RSS values at an unknown position and the fingerprints in the database, and this could lead to big positioning errors due to the instability of the RSS values.

On the other hand, the probabilistic algorithm needs to know the probability distribution of the RSS values from available APs at every RPs. Even though this algorithm can provide good accuracy, it is more complex than the deterministic algorithm, and the positioning performance relies on the computation of likelihood functions. Some popular probabilistic algorithms are the Kalman filter [9], particle filter [30], and hidden Markov models [31]. Using two Kalman filters, Zhuang et al. [32] fused the positional information from the micro-electromechanical system (MEMS) sensors and WiFi fingerprinting to enhance the tracking accuracy. The MEMS sensors helped to reduce the searching space of WiFi fingerprinting by using an extended Kalman filter, and another Kalman filter was applied to smooth the positioning result of WiFi fingerprinting. From two test scenarios, the results showed that the proposed method improved the accuracy by 47% and 28%

with mean errors of 2.2 m and 2.6 m, respectively. In Reference [33], Deng et al. also used the extended Kalman filter in the integrated system, which consisted of pedestrian dead reckoning, WiFi fingerprinting, and special positions in a building (i.e., doors, elevators, escalators, etc.) to reduce the positioning error. From the experiments, it was shown that the mean positioning error was only 1.22 m in a test area of 487.2 m². The two above methods obtained good positioning results; however, they could not completely handle the variation of the RSS values, even though they used different sources (i.e., smartphone sensors and special positions in a building), which would make the systems become more complicated, to calibrate for estimated user's position.

Currently, many latest works use the WiFi fingerprinting technique as the main tool to determine the user's position with the help of machine learning techniques [34–38]. In Reference [37], based on the conventional WiFi fingerprinting, Song et al. proposed the CNNLoc which combined the stacked auto-encoder and convolutional neural network (CNN) to track the users' positions in multi-buildings with multi-floors. The framework required a lot of data preprocessing in the training phase, such as sub-datasets division, rectangle areas creation, cell grids division, etc. As the result, the proposed method obtained a 96.03% accuracy of floor classification with a positioning error of 11.78 m. Moreover, using the deep learning approach, Qin et al. [38] combined the convolutional denoising autoencoder (CDAE) and CNN for the IPS. In the offline phase, the K-means algorithm was used to extract the set of features. During the online phase, the RSS values were put into the CDAE to extract the main features, and then the CNN was used to find out the user's position. The experimental results showed that the mean positioning errors were 1.05 m and 12.4 m for two different datasets. These methods were vulnerable to the changes of RSS values, and they were required to be adjusted to work well with different buildings.

The RSS values collected even from the same AP can fluctuate much due to the changes of environmental conditions, such as the number of working people, the number of electrical devices, the user's body, the period in a day, etc. The aforementioned works have not controlled the instability of the RSS values perfectly. Therefore, we propose a novel WiFi fingerprinting-based method, which utilizes the valued tolerance rough set and decision rules method to classify the user's position among predefined reference points. To the best of our knowledge, applying the VTRS–DR method to solve the indoor positioning challenge has never been recorded before.

3. VTRS–DR Method

Say we have a set of objects, $U = \{u_1, u_2, \dots, u_m\}$ (m is the total number of the fingerprints of RPs), that can be characterized by a set of conditional attributes, $C = \{RSS_1, RSS_2, \dots, RSS_n\}$ (a set of n RSS values collected from n APs at one RP, $n < m$). If we denote a subset of objects in U by a decision attribute $d \notin C$ (with $d \in \{RP_1, RP_2, \dots, RP_N\}$), which is one RP in a set of N RPs, then we define the decision table $DT = (U, C \cup \{d\})$. The decision attribute d partitions set U into N decision classes ($N \leq m$) as $D_l, l = 1, 2, \dots, N$, with each class being one RP in the fingerprinting database. Each decision class is a tolerance class. At the beginning of the VTRS–DR method, the relations between the objects and the attributes can be represented through the decision table, DT , which is given in Table 1. In this table, n conditional attributes (i.e., n RSS values from n APs) $C = \{RSS_1, RSS_2, \dots, RSS_n\}$ are given and each attribute $RSS_j, j = 1, 2, \dots, n$, has its RSS values changing in the interval $[-100, 0]$ (dBm). Here, each object (i.e., one row in Table 1) represents one fingerprint (i.e., a unique set of RSS values) of the given RP, which is a decision class. It is worth noting that, in the decision table, DT , we do not need to sort the RPs by their orders, and the number of appearing times of each decision class (i.e., each RP) can be different which depends on the number of RSS scanning times at each RP.

Table 1. The structure of the decision table.

U	RSS_1 (dBm)	RSS_2 (dBm)	...	RSS_n (dBm)	d
u_1	−51	−54	...	−52	RP_1
u_2	−46	−47	...	−51	RP_2
u_3	−50	−56	...	−53	RP_1
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
u_{m-1}	−76	−43	...	−41	RP_N
u_m	−73	−41	...	−38	RP_N

To explain the VTRS–DR method clearly, we are able to give some basic definitions.

Definition 1. Valued tolerance relation among the objects of set U built on a set of conditional attributes $A \subseteq C$ is denoted by R_A (i.e., $R_A : U \times U \rightarrow [0, 1]$). R_A satisfies two properties: (a) reflexive, $\forall u \in U, R_A(u, u) = 1$; and (b) symmetric, $\forall u, v \in U, R_A(u, v) = R_A(v, u)$.

Definition 1 is used to determine whether the objects of set U satisfy the valued tolerance relation built on a set of conditional attributes, $A \subseteq C$. If a pair of objects has the valued tolerance relation, then these objects may belong to the same tolerance class.

Definition 2. The valued tolerance relation of each attribute RSS_j , in any two objects, $u, v \in U$, is denoted by $R_j(u, v)$ and calculated as follows:

$$R_j(u, v) = \frac{\max(0, \min(RSS_j(u), RSS_j(v)) + k_j - \max(RSS_j(u), RSS_j(v)))}{k_j} \tag{1}$$

where $RSS_j \in A \subseteq C, k_j > 0$ is the discrimination threshold value of the attribute $RSS_j, j = 1, 2, \dots, n$. The discrimination threshold, k , is defined as the threshold of the similarity measure. The value k_j is used to measure the similarity between two values of the attribute RSS_j . From Equation (1), $\forall u, v \in U : R_A(u, v) = 1 \Leftrightarrow RSS_j(u) = RSS_j(v), R_A(u, v) = 0 \Leftrightarrow |RSS_j(u) - RSS_j(v)| \geq k_j$, and $0 < R_A(u, v) < 1 \Leftrightarrow |RSS_j(u) - RSS_j(v)| < k_j$. The value k_j determines whether the two objects are indiscernible (equivalent) with the respect to each attribute in rough set theory.

Definition 2 is used to calculate the valued tolerance relation between a pair of objects in U built on a conditional attribute, $RSS_j \in A \subseteq C$. This definition supports the calculation of the valued tolerance relation between two objects in Definition 3.

Definition 3. The valued tolerance relation between two objects, $u, v \in U$, on a set of conditional attributes, $A \subseteq C$, is denoted by $R_A(u, v)$ and calculated as follows:

$$R_A(u, v) = \min_{RSS_j \in A} (R_j(u, v)) \tag{2}$$

From Equation (2), $\forall u, v \in U : 0 \leq R_A(u, v) \leq 1$, and $R_A(u, v)$ satisfies two properties, (a) and (b), of the valued tolerance relation.

Definition 3 used is to determine the valued tolerance relation between a pair of objects in U built on a set of conditional attributes, $A \subseteq C$. If $R_A(u, v) = 0$, and then two objects, $u, v \in U$, are not related. If $0 < R_A(u, v) \leq 1$, then two objects have a valued tolerance relation. If $R_A(u, v)$ is closer to 1, then two objects are more related. Then the valued tolerance relation matrix is built, $\forall (u, v) \in U \times U : 0 \leq R_A(u, v) \leq 1$, where this matrix is symmetric with the main diagonal is equal to 1. This definition supports Definition 4 to calculate the tolerance granule for each object in U .

Definition 4. A tolerance granule of object $u \in U$ for relation R_A is denoted by $R_A(u)$ and calculated as follows:

$$R_A(u) = \{v \in U : R_A(u, v) > 0\} \quad (3)$$

Definition 4 is used to determine whether the objects in U have the valued tolerance relation with an object, $u \in U$, built on a set of conditional attributes, $A \subseteq C$. Here, $R_A(u)$ is a tolerance granule of u , where it carries a set of objects that has the valued tolerance relation with u (i.e., a set of supporting objects of u). This definition supports Definition 5 to determine the lower and upper approximations of the decision classes (i.e., the tolerance classes) built on a set of conditional attributes, $A \subseteq C$.

Definition 5. With a set of decision classes, $D \subseteq U$, a subset of conditional attributes, $A \subseteq C$, and an object, $u \in U$, the lower approximations and upper approximations of D for A are denoted by D_A and D^A and defined as follows:

$$D_A = \{\forall u \in U : R_A(u) \subseteq D\} \quad (4)$$

$$D^A = \{\forall u \in U : R_A(u) \cap D \neq \emptyset\} \quad (5)$$

Definition 5 used is to determine a lower approximation, D_A , of decision class D , which has tolerance granules, $R_A(u)$, involved in D and an upper approximation, D^A , of D that has tolerance granules, $R_A(u)$, intersected with D . In this way, any objects that belong to D_A are certainly involved in D ; meanwhile, any objects that belong to D^A can be involved in D or not. Therefore, the determination of the lower and upper approximations plays an important role in the performance of the VTRS-DR method since it strongly affects the decision results, as well as the calculation space. The closer the D_A and D^A are to the D , the more accurate the results and the lower the computational cost.

Definition 6. With a set of decision classes, $D \subseteq U$, a subset of conditional attributes, $A \subseteq C$, and an object, $u \in U$, the membership degrees of object $u \in U$ belonging to D_A and D^A are denoted by $\mu_{D_A}(u)$ and $\mu_{D^A}(u)$ and calculated as follows:

$$\mu_{D_A}(u) = \min_{v \in R_A(u)} (\max(1 - R_A(u, v), \hat{v})) \quad (6)$$

$$\mu_{D^A}(u) = \max_{v \in R_A(u)} (\min(R_A(u, v), \hat{v})) \quad (7)$$

where \hat{v} is the membership degree of object v in the set D (i.e., $\hat{v} = \begin{cases} 1, v \in D \\ 0, v \notin D \end{cases}$ or $\hat{v} \in \{0, 1\}$).

Definition 6 is used to calculate the membership degree of an object, $u \in U$, to find out whether it belongs to the lower or upper approximation. The $\mu_{D_A}(u)$ and $\mu_{D^A}(u)$ values are in the interval between 0 and 1; thus, the object, u , will belong to the approximation that is closer to 1. This is an important criterion to determine the decision class, D , that the object, u , belongs to.

Definition 7. With a decision table $DT = (U, C \cup \{d\})$ and a subset of attributes, $A \subseteq C$, a decision rule that describes an object, $u_i \in U$, $i = 1, 2, \dots, m$, is denoted by ρ_i , and defined by the following:

$$\rho_i \stackrel{\text{def}}{=} \bigwedge_{RSS_j \in A} (RSS_j = RSS_j(u_i)) \rightarrow (D = d(u_i)) \quad (8)$$

where $RSS_j(u_i)$ is a value of $RSS_j \in A$ and $(D = d(u_i))$ is a decision class based on decision attribute d of the object u_i .

Definition 7 is used to describe a decision rule, ρ_i , corresponding to an object, $u_i \in U$, $i = 1, 2, \dots, m$. In Equation (8), the left side presents a set of conditional attributes, $A \subseteq C$, and the right side presents a decision attribute, d , corresponding to a decision class that involves the object, u .

Definition 8. With a subset of attributes, $A \subseteq C$, the credibility degree of a decision rule ρ_i , $i = 1, 2, \dots, m$, is denoted by $\mu_A(\rho_i)$ and calculated as follows:

$$\begin{aligned}\mu_A(\rho_i) &= \min_{u \in S_A(\rho_i)} (\max(1 - R_A(\rho_i, u), \mu_{D_A}(u))) \\ &= \min_{u \in R_A(u_i)} (\max(1 - R_A(u_i, u), \mu_{D_A}(u)))\end{aligned}\quad (9)$$

Definition 8 is to calculate the credibility degree of a decision rule, ρ_i , $i = 1, 2, \dots, m$. The $\mu_A(\rho_i)$ value is in the interval between 0 and 1. If this value is closer to 1, then it is reliable that the object u_i is involved in the decision class D . This is an important criterion for classifying a new object into a decision class.

Definition 9. With a subset of attributes, $A \subseteq C$, a support object set of a decision rule ρ_i , $i = 1, 2, \dots, m$, is denoted by $S_A(\rho_i)$ and calculated as follows:

$$S_A(\rho_i) = \{u \in U : R_A(\rho_i, u) > 0\} = \{u \in U : R_A(u_i, u) > 0\} = R_A(u_i) \quad (10)$$

In Equation (10), $S_A(\rho_i)$ is the tolerance granule of rule ρ_i for relation s_A . It means, on binary relation $\{\rho_i\}_{i=1}^m \times U$, and with a set of conditional attributes, $A \subseteq C$, we build a valued tolerance relation, $s_A : \{\rho_i\}_{i=1}^m \times U \rightarrow [0, 1]$, such that $s_A(\rho_i, u) > 0$, where $s_A(\rho_i, u)$ is the support degree of rule ρ_i for object $u \in U$. Object u is similar to some extent to the conditional part of the rule ρ_i on the set of conditional attributes, $A \subseteq C$. Moreover, s_A is a valued tolerance relation defined exactly as the relation R_A . Thus, $s_A(\rho_i, u)$ is calculated as in Equation (3), i.e., $\forall u \in U$ and $i = 1, 2, \dots, m$, $s_A(\rho_i, u) \equiv R_A(\rho_i, u) \equiv R_A(u_i, u)$, where u_i is an object of the rule ρ_i .

Definition 9 is to determine the $S_A(\rho_i)$, a support object set of a decision rule, ρ_i , corresponding to an object, u_i , $i = 1, 2, \dots, m$. The $S_A(\rho_i)$ of ρ_i is equivalent to a tolerance granule, $R_A(u_i)$, of the object u_i . If many decision classes have the same credibility degrees, then it is difficult to decide which class the object u_i belongs to; thus, the support object set of each class will help to solve the problem of classification. This value plays an important role in classifying a new object into a decision class.

4. The Proposed Positioning Method

In this section, to apply the VTRS-DR method to the WiFi fingerprinting method, we recount how we used the VTRS-DR method with a set of conditional attributes, $A = C$, meaning that the whole set of C (i.e., the whole APs) in the decision table, DT , was applied. Figure 1 shows the overall structure of the proposed method. The method is applied not only in the offline phase but also in the online phase. For the offline phase, at each reference point, RP_l , $l = 1, 2, \dots, N$, a large number of RSS values from n APs is collected. Then the original RSS fingerprinting database is created, as shown in Table 2. The structure of this database is presented as follows: {Position, Coordinate, RSS values from APs}. To make this database similar to the decision table, DT , in Table 1, two parts {Position, RSS values from APs} in the original database are copied, and the index of each row is added as the first column to mark the objects. Finally, Algorithm 1 is applied to the DT to make the fingerprinting decision rules database. The structure of the fingerprinting decision rules database is shown in Table 3, with the format as follows: {Rule, Conditions of rule, Decision attribute, Support components}. Here, the support components include two values: the credibility degree and the support object set value. For the online phase, based on the fingerprinting decision rules database and measured RSS values $\theta = (\theta_1, \theta_2, \dots, \theta_n)$ received from one unknown position, the user's position is obtained via Algorithm 2. Algorithm 2 chooses the best RP candidate as the user's position among the whole RPs by considering the valued tolerance relation, the credibility degree, the support object set value, and the Euclidean distance. One example of the proposed method can be seen in Appendix A.

Algorithm 1: The Construction of a Fingerprinting Decision Rules Database for IPS

Input: The decision table DT .

Output: The fingerprinting decision rules database.

Step 1. Create the valued tolerance relation matrix from relations of all pairs of two objects (two rows) in DT : $R_A(u, v), \forall u, v \in U$ (Equation (1) and Equation (2)).

Step 2. Determine the tolerance granule, $R_A(u), \forall u \in U$ (Equation (3)).

Step 3. Determine the set of decision classes D (i.e., a total number of RPs) of decision table, DT , based on the decision attribute, d . For each decision class, D_l , calculate the lower approximations, $(D_l)_A$ and the upper approximations, $(D_l)^A, l = 1, 2, \dots, N$ (Equation (4) and Equation (5)).

Step 4. Calculate the membership degrees, $\mu_{(D_l)_A}(u)$ and $\mu_{(D_l)^A}(u)$, of object u , corresponding to $(D_l)_A$ and $(D_l)^A, \forall u \in U, \forall D_l \in \{D_1, D_2, \dots, D_N\} \equiv \{RP_1, RP_2, \dots, RP_N\}$ (Equation (6) and Equation (7)).

Step 5. Define the decision rule, ρ_i , that describes the object, $u_i \in U$, on the set of attributes $A, i = 1, 2, \dots, m$ (Equation (8)).

Step 6. Calculate the credibility degree, $\mu_A(\rho_i)$, of a decision rule, $\rho_i, i = 1, 2, \dots, m$ (Equation (9)).

Step 7. Define the support object set, $S_A(\rho_i)$, of a decision rule, ρ_i , for the set of attributes $A, i = 1, 2, \dots, m$ (Equation (10)).

Step 8. Create the fingerprinting decision rules database.

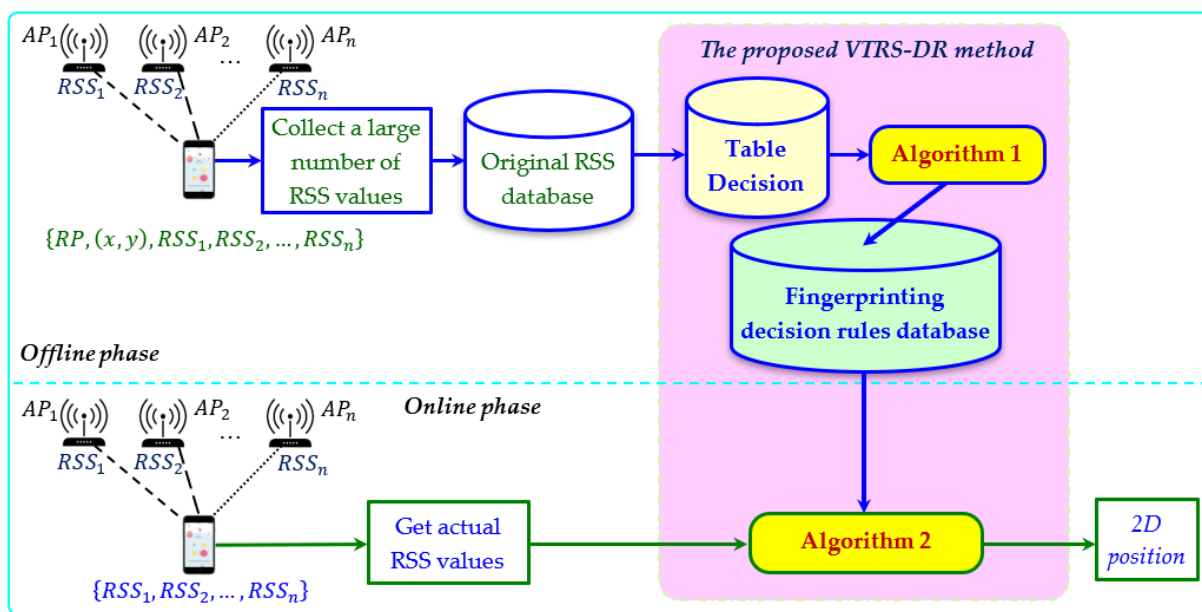


Figure 1. The overall structure of the proposed method.

Table 2. The original RSS database.

Position	Coordinates	RSS_1 (dBm)	RSS_2 (dBm)	...	RSS_n (dBm)	
RP_1	(x_1, y_1)	-51	-54	...	-52	} K fingerprints
RP_1	(x_1, y_1)	-50	-56	...	-53	
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	
RP_1	(x_1, y_1)	-54	-58	...	-55	} K fingerprints
RP_2	(x_2, y_2)	-46	-47	...	-51	
RP_2	(x_2, y_2)	-44	-49	...	-50	
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	} K fingerprints
RP_2	(x_2, y_2)	-48	-45	...	-49	
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	
RP_N	(x_N, y_N)	-76	-43	...	-41	} K fingerprints
RP_N	(x_N, y_N)	-73	-41	...	-38	
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	
RP_N	(x_N, y_N)	-75	-44	...	-39	

Table 3. The structure of the fingerprinting decision rules database.

Rule	Conditions of Rule				Decision Attribute	Support Components		
	RSS_1 (dBm)	RSS_2 (dBm)	...	RSS_n (dBm)	d (Position)	Credibility Degree (μ_A)	Support Object Set (S_A)	
ρ_1	-51	-54	...	-52	RP_1	$\mu_A(\rho_1)$	$S_A(\rho_1)$	} K rules
ρ_2	-50	-56	...	-53	RP_1	$\mu_A(\rho_2)$	$S_A(\rho_2)$	
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	
ρ_K	-54	-58	...	-55	RP_1	$\mu_A(\rho_K)$	$S_A(\rho_K)$	} K rules
ρ_{K+1}	-46	-47	...	-51	RP_2	$\mu_A(\rho_{K+1})$	$S_A(\rho_{K+1})$	
ρ_{K+2}	-44	-49	...	-50	RP_2	$\mu_A(\rho_{K+2})$	$S_A(\rho_{K+2})$	
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	} K rules
ρ_{2K}	-48	-45	...	-49	RP_2	$\mu_A(\rho_{2K})$	$S_A(\rho_{2K})$	
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	
$\rho_{(N-1)K+1}$	-76	-43	...	-41	RP_N	$\mu_A(\rho_{(N-1)K+1})$	$S_A(\rho_{(N-1)K+1})$	} K rules
$\rho_{(N-1)K+2}$	-73	-41	...	-38	RP_N	$\mu_A(\rho_{(N-1)K+2})$	$S_A(\rho_{(N-1)K+2})$	
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	
ρ_{NK}	-75	-44	...	-39	RP_N	$\mu_A(\rho_{NK})$	$S_A(\rho_{NK})$	

Algorithm 2: The Advanced Positioning Algorithm Based on the Proposed VTRS–DR Method

Input: Set of measured RSS values, θ , fingerprinting decision rules database.

Output: Position of the user.

Step 1. Calculate the valued tolerance relation between θ to conditions of rule ρ_i : $R_A(\theta, \rho_i)$, $i = 1, 2, \dots, m$ (i.e., $R_A(\theta, \rho_i) = \min_{RSS_j \in A} (R_j(\theta, \rho_i))$, as in Equation (2)).

Step 2. Determine the smallest value $\mu_{\rho_i}(\theta)$ between the valued tolerance relation $R_A(\theta, \rho_i)$ and the credibility degree of rule ρ_i : $\mu_{\rho_i}(\theta) = \min(R_A(\theta, \rho_i), \mu_A(\rho_i))$, $i = 1, 2, \dots, m$.

Step 3. Calculate the credibility degree of θ to each decision class D_l (i.e., one RP), $l = 1, 2, \dots, N$: $\mu_{D_l}(\theta) = \max_{\rho_i \in R(D_l)} (\mu_{\rho_i}(\theta))$, where $R(D_l)$ is the set of all decision rules that belongs to D_l and having $\mu_{\rho_i}(\theta) > 0$, $i = 1, 2, \dots, m$.

Step 4. Choose the class D_l that maximizes the credibility degree $\mu_{D_l}(\theta)$. If $\exists! D_l : \mu_{D_l}(\theta)$ is at the maximum, then choose this D_l as the user's position.

Step 5. If the same maximum credibility degrees exist on different L ($L < N$) decision classes, $\{D_{l_1}, D_{l_2}, \dots, D_{l_L}\}$, then calculate the relative supports of rules in $\{D_{l_1}, D_{l_2}, \dots, D_{l_L}\}$, as denoted by $Supp(\rho_i)$. Note that $Supp(\rho_i)$ is a ratio of the number of the support object set $S_A(\rho_i)$ to the total number of the set of all objects in any decision class which has the rule ρ_i .

Step 6. Calculate the aggregated support of θ as $Supp_{D_{l_p}}(\theta) = \max_{\rho_i \in R(D_{l_p})} (Supp(\rho_i))$, $p = 1, 2, \dots, L$.

Step 7. Choose the D_{l_p} that maximizes the aggregated support $Supp_{D_{l_p}}(\theta)$. If $\exists! D_{l_p} : Supp_{D_{l_p}}(\theta)$ is at the maximum, then choose this D_{l_p} as the user's position.

Step 8. If the same maximum aggregated support values, $Supp_{D_{l_p}}(\theta)$, exist on many decision classes, then use the Euclidean distance to choose the decision class D_{l_p} which has the minimum distance between the set of measured RSS values θ and the mean of RSS values of that decision class; then choose that class (i.e., one RP) as the user's position.

5. Experimental Results

5.1. Experimental Setup

To validate the efficiency and accuracy of the proposed method, we used the dataset introduced by Duong-Bao et al. [39]. The dataset was created in an office room at Hunan University, China, over four months. The room had an area of $9.0 \times 6.5 \text{ m}^2$. Five APs were installed at different positions, and each one was fixed at 1.1 m to 1.6 m from the ground, as can be seen in Figure 2. The smartphone played a role as a client to send the scanned RSS values to the server (i.e., the laptop). This server was responsible for storing the RSS values, creating the radio map, and classifying the user's position among predefined RPs. Figure 3 shows the positions of the APs, as well as the RPs in a two-dimensional coordinate. From this figure, there existed 205 RPs in the room, and the grid spacing between two adjacent RPs was 0.5 m. The starting point (RP0) was marked at the entrance door on the top left of the figure, and the other RPs were marked orderly from the left side to the right side, and from the top to the bottom. The last point (RP204) ended at the bottom right, where it was nearby the AP3.

In the offline phase, a subject holding a smartphone in front of his body stood at each determined RP to collect the RSS values from the five available APs. For each RP, the subject scanned the RSS values 100 times in different directions, meaning that there were 20,500 scanning times for 205 RPs. In this phase, various environmental conditions (i.e., the density of people and electrical devices, the direction of the user, the period of a day, etc.) were considered to create a noisy and complex fingerprinting database. In the online phase, the RSS values were collected in two separate test cases which were set up with different environmental conditions. Table 4 shows the conditions for the RSS collection of the two cases. For each case, the subject stood on each RP and used the smartphone to scan the RSS values once, meaning that the subject collected the RSS values 205 times over 205 RPs (i.e., 410 times in two cases). More detailed descriptions of the dataset can be found in Reference [31].



Figure 2. The office room for data collection.

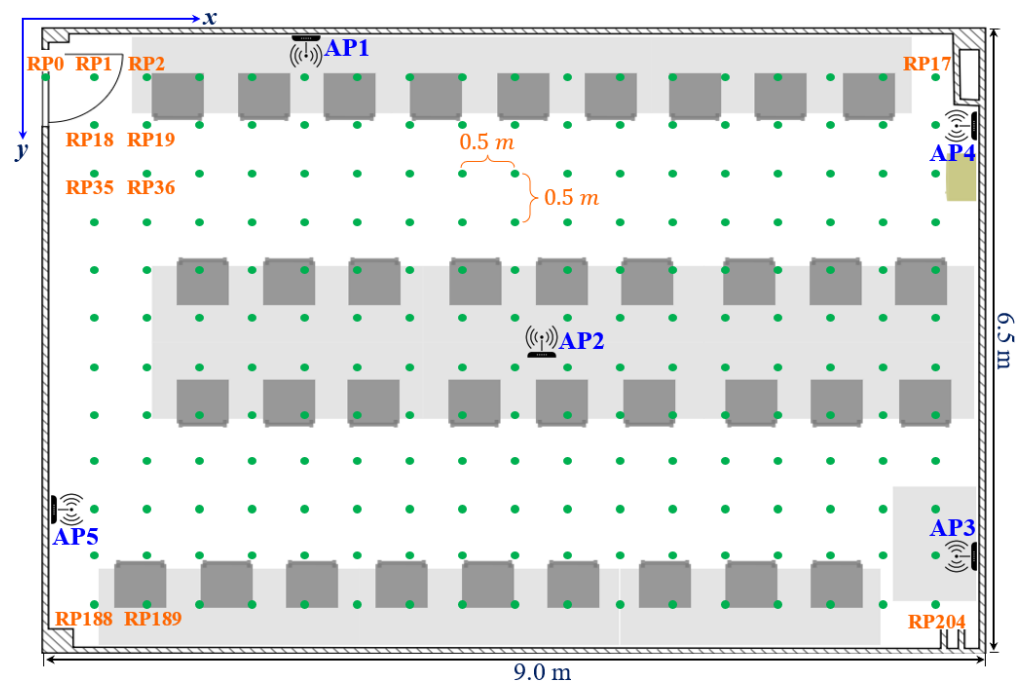


Figure 3. The positions of five APs and 205 APs in the office room.

Table 4. The difference in environmental conditions in Case 1 and Case 2.

Conditions	Case 1	Case 2
Density of people	1 to 9	6 to 13
Density of electrical devices	11	20
Temperature	Cool	Warm
Height from the ground	1.3 m	1.3 m
Subject direction	Random	Random

5.2. Experimental Results

From the aforementioned dataset, in this experiment, the fingerprinting decision rules database was created from the original RSS fingerprinting database, which involves the RSS values collected from 205 RPs in the offline phase. Therefore, the training data used to build the new database include 20,500 objects corresponding to 20,500 samples (i.e., 20,500 rows) and five conditional attributes corresponding to five APs (i.e., five columns). At each RP position, the subject collected the RSS values 100 times; thus, there are 100 rules for each class (i.e., each RP), which also means that, for each RP, there are 100 collected fingerprints (i.e., set of RSS values). The testing data used the two test cases that were collected in the online phase of the above dataset to classify the user's position. For each test case, the RSS values from the five APs were collected at each RP once, meaning there were 205 sets of RSS values collected over 205 RPs used for performance evaluation.

The discrimination thresholds, k , corresponding to the number of conditional attributes are shown in Table 5. To the best of our knowledge, the k values are mostly chosen from the trial-and-error method, meaning that different sets of k values were put into the proposed method and repeatedly tested to find out the best k values. We chose the discrimination threshold values, k_j , for each attribute RSS_j , $j = 1, 2, \dots, n$, to measure the similarity among the objects and calculate the experimental results to find out which set of k_j can help the VTRS-DR method to achieve the best result. With each set of k_j , the VTRS-DR method builds a fingerprinting decision rules database. We used the database with 20,500 rows to test with the fingerprinting decision rules database corresponding to each k . From the classification results corresponding to different sets of k values, we then chose the set that maximizes the result.

Table 5. The discrimination threshold values, k .

Attribute	RSS_1	RSS_2	RSS_3	RSS_4	RSS_5
k_j	1.97	1.98	1.96	1.95	1.99

To evaluate the performance of our proposed method, we compared it with the NN-based methods (i.e., the 1-NN and WKNN (K is chosen as 3)) [17] and the random statistical (RS) method [14]. Note that the NN-based methods use the original fingerprinting database and the RS method uses the standardized fingerprinting database; meanwhile, the proposed method uses the fingerprinting decision rule database when matching with the measured RSS values in the online phase. For the compared methods, in the offline phase, the fingerprint of one RP is presented as a set of means of collected RSS values (i.e., 100 RSS values for each available AP at that RP). In the online phase, the NN-based and RS methods calculate the distance between the measured RSS values with the fingerprints of the whole RPs to find out the user's position. The Euclidean distance was used for the two NN-based methods, while the Mahalanobis distance was used for the RS method, respectively. The mean error in the experiment is calculated as follows:

$$\Delta_{ME} = \frac{\sum_{l=1}^N E_l}{N} \quad (11)$$

where Δ_{ME} is the mean error (m), N is the number of RPs used in the online phase ($N = 205$), and E_l is the Euclidean distance between RP_l and its classified position.

Figure 4 shows the results of positioning errors of the four methods for the dataset of Case 1. It can be seen from the figure that the VTRS-DR method achieves the best results, as it can mostly find out the exact user's positions among the predefined RPs (i.e., the same positions as the RPs where the user stood), which means no error (0 m). In Figure 4, there are only four positions (the 15th, 44th, 86th, and 162nd positions) where the proposed method obtains the wrong results, while the 1-NN and RS methods obtain 93 and 113 wrong results. The positioning errors corresponding to the four wrong estimated positions are 1 m, 3.35 m, 1.80 m, and 0.71 m, respectively. Comparing the mean positioning

errors, the 1-NN, WKNN, and RS methods achieve errors around 1 m, which are 1.13 m for 1-NN, 1.17 m for WKNN, and 0.94 m for RS, respectively. Meanwhile, the mean errors of the proposed method are far better than others with only 0.033 m. A deeper statistical comparison of error positions is pointed out in Table 6. In this table, we can see that the WKNN method has the smallest minimum positioning error, with only 0.05 m, while the maximum error of the VTRS-DR method is only 3.35 m, which is approximately half of that of other methods, with 8.02, 7.03 m, and 6.73 m, respectively. However, the mean positioning error at wrong classified positions of the proposed method is not the smallest one (1.71 m with the standard deviation of 1.03 m), as it is higher than the WKNN mean error of only 1.17 m but lower than the two others, 2.05 m and 2.06 m. The results show the superiority of the VTRS-DR method to other methods for the dataset of Case 1 since it improves the classified accuracy from the above methods by 53.17% and 43.42%. The WKNN method cannot provide the exact positions where the user stood since this method calculates the mean positions from the three nearest candidates.

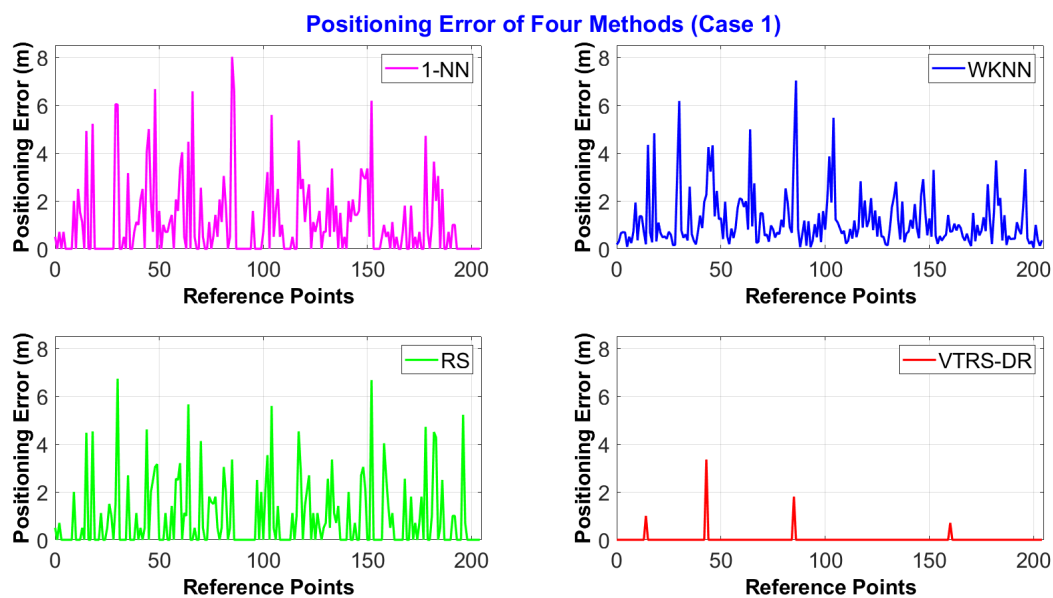


Figure 4. The positioning error distance of four methods via 205 RPs in Case 1.

Table 6. Statistical comparison of four methods in Case 1 at error positions.

	1-NN	WKNN	RS	VTRS-DR
No. of Errors	113	205	93	4
Max (m)	8.02	7.03	6.73	3.35
Min (m)	0.50	0.05	0.50	0.71
Mean (m)	2.05	1.17	2.06	1.71
Stdev (m)	1.69	1.15	1.52	1.03

Figure 5 displays the positioning errors of the four methods for the dataset of Case 2, where the environmental conditions were set up as being more complicated, such as increasing the number of working people or increasing the number of electrical devices. Overall, even though the testing condition is more complex, the proposed method is still better than others since it still maintains a very small number of wrong estimated positions with only four errors (the 46th, 62nd, 86th, and 99th positions) compared to the 1-NN (116) and RS (108) methods, meaning that it reduces the classification errors by 54.64% and 50.73%, respectively. The positioning errors corresponding to the four wrong estimated positions are 1.58 m, 3.16 m, 1.80 m, and 1.41 m, respectively. The mean positioning error of

the proposed method is the best one with only 0.039 m, while the others' errors are around 1.2 m. Table 7 shows the further statistical analysis of the four methods in Case 2 at error positions. From this table, the proposed method still achieves the smallest maximum error with 3.16 m, but the mean positioning error of the WKNN method is higher than others with 1.26 m.

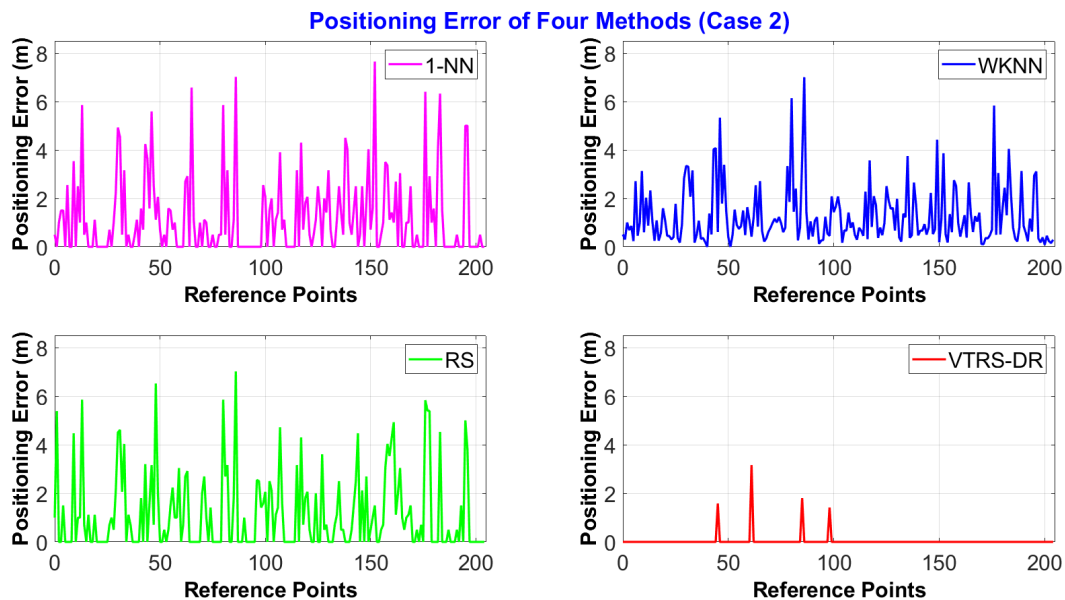


Figure 5. The positioning error distance of four methods via 205 RPs in Case 2.

Table 7. Statistical comparison of four methods in Case 2 at error positions.

	1-NN	WKNN	RS	VTRS-DR
No. of Errors	116	205	108	4
Max (m)	7.65	7.00	7.02	3.16
Min (m)	0.50	0.001	0.50	1.41
Mean (m)	2.10	1.26	2.21	1.99
Stdev (m)	1.67	1.20	1.62	0.69

Moreover, the cumulative position error distributions in the two test cases are shown in Figure 6. For Case 1 (a), at the 60th percentile, the positioning error of the proposed method is 0 m, and at the 90th percentile, the error is still 0 m. The 0 m error continues until the 98th percentile. At the same percentiles, the errors of the NN, the WKNN, and RS methods are 0.68 m, 0.48 m, and 0.93 m at the 60th percentile and 3.24 m, 3.02 m, and 2.61 m at the 90th percentile, respectively. From this result, the VTRS-DR method certainly achieves much better accuracy compared to others. For Case 2 (b), the same phenomenon for the positioning errors among four methods also happens where the error of the proposed method is greatly smaller than the results of the other methods at the same percentiles. From both test cases, the VTRS-DR method outperforms the performance of the other methods from the point of view of classification since it can help to correctly classify the unknown position among the predefined RPs. The major reason for the extraordinary results of the proposed method is that it uses many classification levels (i.e., calculating the valued tolerance relation, the credibility degree, the support object set value, and the Euclidean distance).

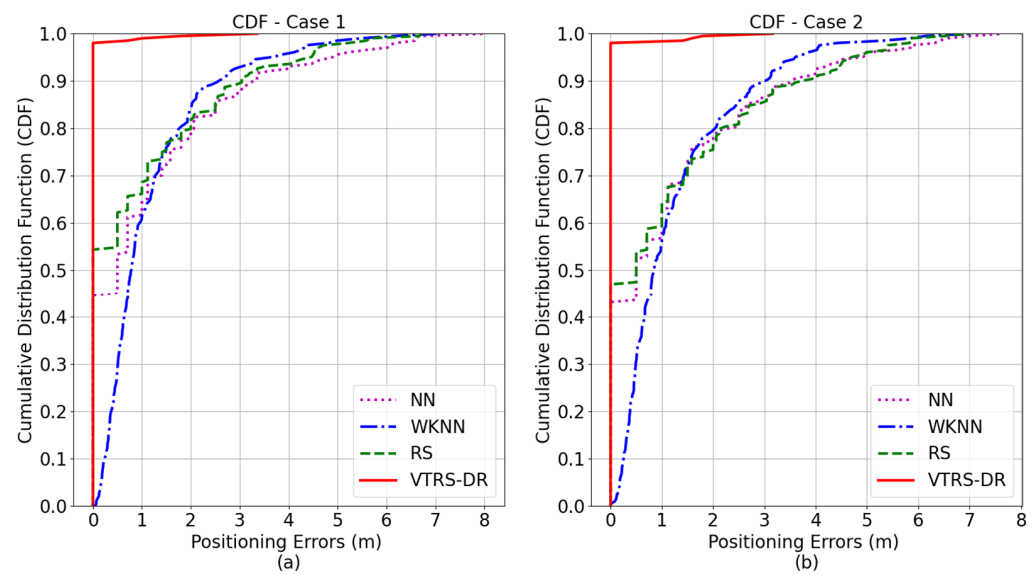


Figure 6. Cumulative position error distributions of the four methods in (a) Case 1 and (b) Case 2.

The computational cost of the four implemented methods is analyzed by calculating the running time of each method on MATLAB (R2019a). We used a Dell Vostro laptop which was equipped with an Intel(R) Core (TM) i7-8550U CPU @1.80 GHz processor, 8 GB RAM, and 128 GB SSD. The operating system was Microsoft Windows 10 Professional Edition (64-bit). The computational cost for creating the fingerprinting decision rules database in the offline phase is approximately 96.62 min, and the time for creating the valued tolerance relation matrix obtains 89.34% of the total running time. Furthermore, Figure 7 shows the running time of the four algorithms over 205 RPs in Case 1 to classify the user's position in the online phase. The figure revealed that the 1-NN becomes the smallest average cost of about 0.0002 s while the VTRS-DR method obtains the biggest average cost of about 0.12 s. Concurrently, the WKNN and RS methods obtain the average values of 0.0077 and 0.0155 s, respectively. From the above costs in two phases, we can easily see that the proposed method will take lots of time to build the fingerprinting decision rules databases, especially the valued tolerance relation matrix, but it does not take much time to estimate the user's position in the online phase. This means that the proposed method can be used in the real-time scenario, while the database creation can be performed on the server readily.

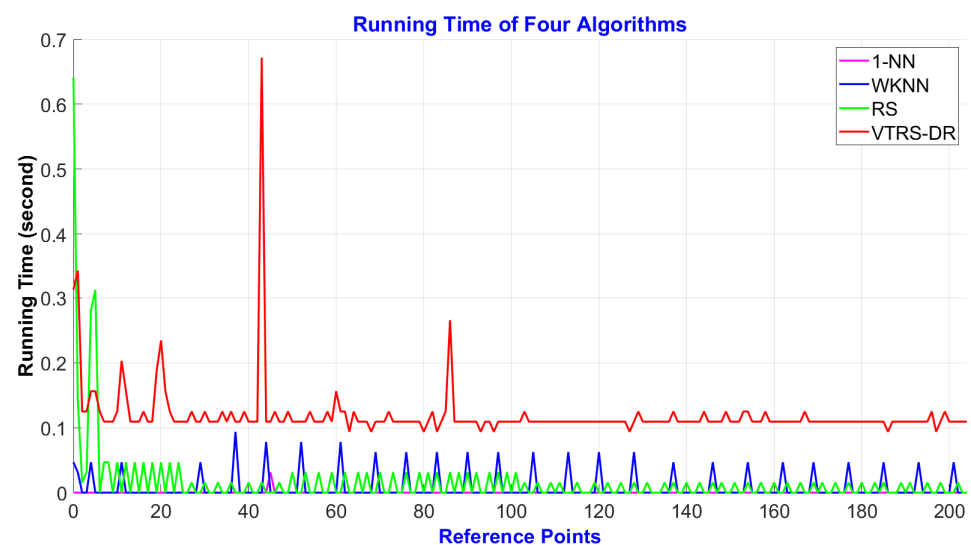


Figure 7. The running time of four methods via 205 RPs in Case 1.

Furthermore, to evaluate the robustness of the proposed method, we intentionally reduced the number of APs to four and three APs instead of using the default five APs. Since there are many possible sets of using four or three APs, we used the combination method to consider the whole possible cases. Therefore, if we consider three APs, the number of possible cases is $C_5^3 = 10$, and with four APs, this number is $C_5^4 = 5$, respectively. The positioning results of the proposed method in two test cases were also compared to the 1-NN, WKNN, and RS methods. Tables 8 and 9 show the mean positioning errors among the four methods in two cases with different combinations of four APs. For example, the notation {1, 2, 3, 4} in the tables represents the combination of four APs from 1 to 4. From both tables, even though the mean positioning errors of all cases increase, the VTRS-DR method still shows the best performance. The error of the proposed method is by far much lower than the others.

Table 8. Mean errors of using four APs in Case 1.

Case 1 (4 APs)	{1, 2, 3, 4}	{1, 2, 3, 5}	{1, 2, 4, 5}	{1, 3, 4, 5}	{2, 3, 4, 5}	Mean (m)
1-NN	1.374	1.433	1.411	1.471	1.390	1.416
WKNN	1.352	1.346	1.532	1.371	1.514	1.423
RS	1.588	1.350	1.407	1.462	1.636	1.489
VTRS-DR	0.210	0.184	0.315	0.227	0.334	0.254

Table 9. Mean errors of using four APs in Case 2.

Case 2 (4 APs)	{1, 2, 3, 4}	{1, 2, 3, 5}	{1, 2, 4, 5}	{1, 3, 4, 5}	{2, 3, 4, 5}	Mean (m)
1-NN	1.527	1.536	1.732	1.473	1.520	1.558
WKNN	1.599	1.537	1.735	1.496	1.563	1.586
RS	1.588	1.538	1.663	1.477	1.650	1.583
VTRS-DR	0.183	0.116	0.237	0.191	0.174	0.180

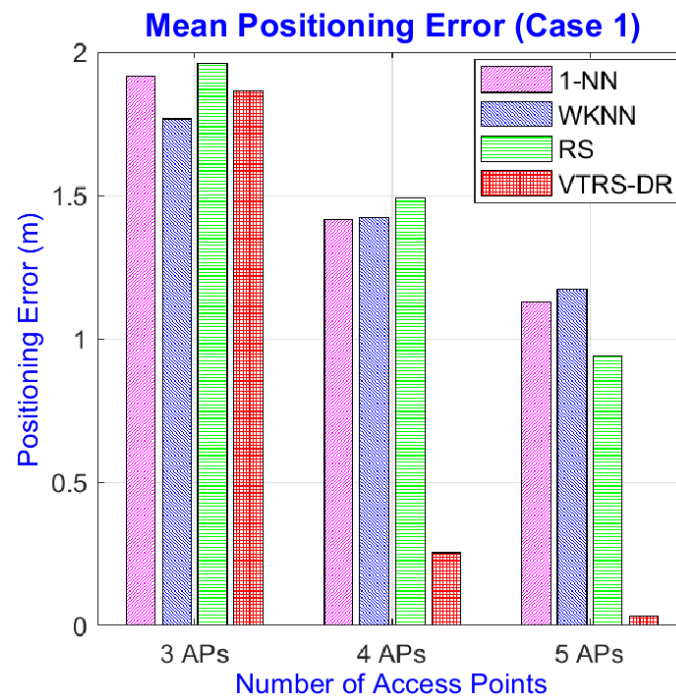
However, this is not right when using only three APs in two test cases. The mean positioning errors can be seen in Tables 10 and 11. The results show that the VTRS-DR method cannot outperform the positioning results of other methods; even some combinations' results of the proposed method are much worse than others, for example, the cases of {1, 3, 4} in both cases. The reason can come from the characteristic of the VTRS-DR method that it may require more than a certain number of attributes (i.e., the number of APs) to produce good results. This problem, however, might not occur in practical scenes due to the widespread of APs indoors, especially in urban areas. Figures 8 and 9 display the comparisons of the mean of mean errors of the four methods in three different numbers of APs. From these figures, we can see that the mean of mean errors of the VTRS-DR methods is much better than others in cases using five and four APs, but for three APs, it is slightly better than others in Case 2 but somewhat worse than WKNN method in Case 1 (i.e., 1.766 m for WKNN and 1.865 m for VTRS-DR). Meanwhile, the RS method obtains the worst error in Case 1 with 1.959 m, and the 1-NN method gets the worst error in Case 2 with 2.103 m. The accuracy of the proposed method in Case 2 is better than in Case 1; we think that this is because the RSS values in Case 1 are more stable at any RPs than in Case 2 due to their simpler setup; thus, there may exist many valued tolerance relations that have the same values, and this makes it hard to classify the decision classes clearly. Therefore, the proposed method can work well in complicated environments, which are more popular in real life than stable environments are, such as Case 1.

Table 10. Mean errors of using three APs in Case 1.

Case 1 (3 APs)	{1, 2, 3}	{1, 2, 4}	{1, 2, 5}	{1, 3, 4}	{1, 3, 5}	{1, 4, 5}	{2, 3, 4}	{2, 3, 5}	{2, 4, 5}	{3, 4, 5}	Mean (m)
1-NN	1.845	1.858	2.000	1.999	1.713	1.902	2.011	1.716	2.230	1.878	1.915
WKNN	1.679	1.843	1.778	1.741	1.532	1.721	1.864	1.715	2.050	1.732	1.766
RS	1.785	1.930	1.879	1.996	1.643	1.821	2.138	1.977	2.338	2.089	1.959
VTRS- DR	1.478	1.848	1.650	3.486	1.237	1.810	2.004	1.444	2.024	1.664	1.865

Table 11. Mean errors of using three APs in Case 2.

Case 2 (3 APs)	{1, 2, 3}	{1, 2, 4}	{1, 2, 5}	{1, 3, 4}	{1, 3, 5}	{1, 4, 5}	{2, 3, 4}	{2, 3, 5}	{2, 4, 5}	{3, 4, 5}	Mean (m)
1-NN	1.912	2.120	2.294	2.133	1.993	2.125	2.159	2.001	2.418	1.873	2.103
WKNN	1.793	1.975	2.028	1.798	1.740	2.018	1.934	1.929	2.234	1.749	1.920
RS	1.821	2.112	2.292	1.954	1.925	2.137	2.165	2.058	2.521	1.947	2.093
VTRS- DR	1.447	1.828	1.735	3.390	1.461	1.733	1.921	1.631	1.920	1.597	1.866

**Figure 8.** Mean of mean errors of four methods with different numbers of APs in Case 1.

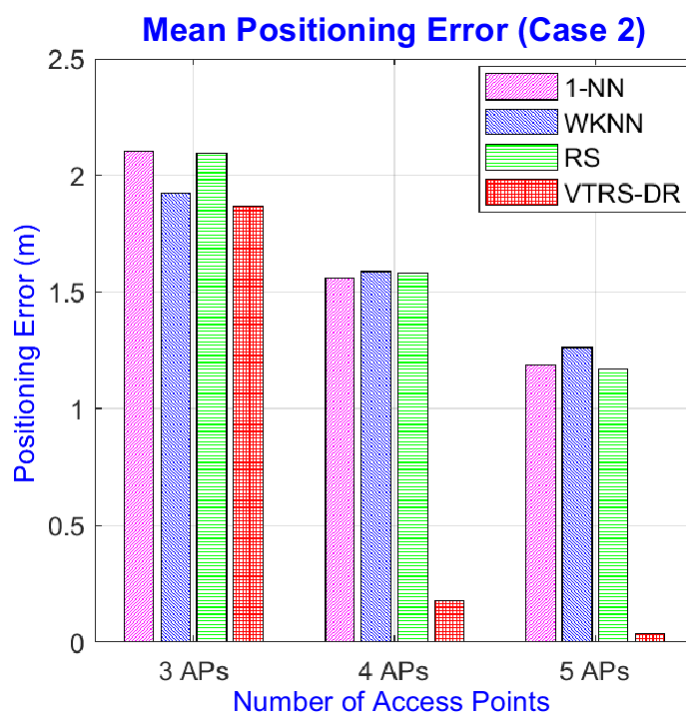


Figure 9. Mean of mean errors of four methods with different numbers of APs in Case 2.

6. Conclusions

In this paper, a novel WiFi fingerprinting positioning method based on the valued tolerance and decision rules method was proposed. The proposed method is implemented not only in the offline phase but also in the online phase. In the offline phase, the fingerprinting decision rules database is built from the decision table based on the conventional RSS fingerprinting database. The created database consists of different components, such as the credibility degree and support object set, which are utilized to support the classification of the measured RSS values in the online phase to the precise class. This helps to increase the classification performance. The evaluation shows that the proposed method outperforms the other methods for both the datasets of the two cases which were set up with different environmental conditions, although the test data were collected in a small number (only two sets) of RSS samples on the exact position of each RP rather than the random positions in the target area. The wrong classification rate of the proposed method is very low, which reduced the errors from the other methods from 43.42% to 54.64%. However, the proposed method also has some limitations. The first one is high complexity. Even though it can achieve extraordinary positioning results, the computational complexity makes it difficult to be used in real-time since it needs to compute the relation matrix in the offline phase, as well as the complicated components, such as the credibility degree and support object set values in the online phase. The second limitation is that the performance of the proposed method will greatly degrade when the number of available APs is reduced. For future work, we want to evaluate the performance of this method in bigger areas, such as buildings with multiple floors and various available APs that consist of more complicated environmental conditions. Furthermore, we also want to investigate how to optimize the number, as well as the setup positions, of APs in a target area.

Author Contributions: Conceptualization, N.D.-B. and J.H.; methodology, N.D.-B.; software, N.D.-B.; validation, L.N.T., K.N.-H. and S.-W.L.; formal analysis, N.D.-B. and K.N.-H.; data curation, N.D.-B.; writing—original draft preparation, N.D.-B.; writing—review and editing, L.N.T., K.N.-H. and S.-W.L.; supervision, J.H.; project administration, J.H.; funding acquisition, J.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the National Natural Science Foundation of China (NSFC) (61775054) and by the National Natural Science Foundation of Hunan Province (grant no. 2020JJ4210).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data for this article can be found in the online version at <https://github.com/luongnt1983/IPS> (accessed on 15 June 2022).

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

Appendix A

In this part, we provide an example of how to use the VTRS–DR method to determine the user’s position. Suppose we set up five APs and there are ten predefined RPs, $RP_1, RP_2, \dots, RP_{10}$, in a test area. At each RP, we collected the RSS values twice, as in Table A1. The decision table, DT , was created from Table A1, as shown in Table A2.

Table A1. The original RSS database proposed for the example.

Position	Coordinates	RSS ₁ (dBm)	RSS ₂ (dBm)	RSS ₃ (dBm)	RSS ₄ (dBm)	RSS ₅ (dBm)
RP_1	(x_1, y_1)	−57	−48	−64	−45	−52
RP_1	(x_1, y_1)	−58	−49	−63	−48	−53
RP_2	(x_2, y_2)	−57	−43	−65	−51	−53
RP_2	(x_2, y_2)	−58	−42	−67	−51	−55
RP_3	(x_3, y_3)	−55	−54	−63	−52	−57
RP_3	(x_3, y_3)	−56	−51	−62	−52	−58
RP_4	(x_4, y_4)	−40	−50	−57	−52	−53
RP_4	(x_4, y_4)	−43	−51	−61	−53	−53
RP_5	(x_5, y_5)	−41	−53	−62	−39	−50
RP_5	(x_5, y_5)	−42	−53	−63	−39	−49
RP_6	(x_6, y_6)	−50	−49	−61	−53	−50
RP_6	(x_6, y_6)	−50	−50	−61	−53	−51
RP_7	(x_7, y_7)	−50	−56	−65	−45	−48
RP_7	(x_7, y_7)	−51	−56	−69	−46	−49
RP_8	(x_8, y_8)	−57	−46	−65	−48	−56
RP_8	(x_8, y_8)	−58	−47	−64	−50	−55
RP_9	(x_9, y_9)	−59	−49	−63	−47	−56
RP_9	(x_9, y_9)	−60	−51	−64	−47	−54
RP_{10}	(x_{10}, y_{10})	−67	−51	−66	−51	−42
RP_{10}	(x_{10}, y_{10})	−66	−53	−66	−51	−43

Table A2. The structure of the decision table created from the original RSS database.

U	RSS_1 (dBm)	RSS_2 (dBm)	RSS_3 (dBm)	RSS_4 (dBm)	RSS_5 (dBm)	d (Position)
u_1	−57	−48	−64	−45	−52	RP_1
u_2	−58	−49	−63	−48	−53	RP_1
u_3	−57	−43	−65	−51	−53	RP_2
u_4	−58	−42	−67	−51	−55	RP_2
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
u_{19}	−67	−51	−66	−51	−42	RP_{10}
u_{20}	−66	−53	−66	−51	−43	RP_{10}

From Definition 1, the objects that have the valued tolerance relation can belong to the same tolerance class (i.e., decision class). Thus, we calculated the valued tolerance relation for all pairs of objects in the decision table, DT (Definition 3). To keep the generality, we choose $A = C$ and $u, v \in U$. The valued tolerance relation was calculated from Equations (1) and (2). In this example, we chose k_j corresponding to each attribute $RSS_j, j = 1, 2, \dots, 5$, based on the standard normal distribution. Then we applied Definition 3 to calculate the valued tolerance relation for all pairs of objects in the decision table, DT (Table 2). For example, we calculated the relation between the two objects u_3 and u_4 .

$$\begin{aligned}
 R_A(u_3, u_4) &= \min\{R_1(u_3, u_4), R_2(u_3, u_4), R_3(u_3, u_4), R_4(u_3, u_4), R_5(u_3, u_4)\} \\
 &= \min\{0.868, 0.725, 0.213, 1.0, 0.517\} = 0.213
 \end{aligned}$$

By applying the same way to all pairs, we calculated the valued tolerance relation for 20 objects in Table A2 and finally obtained the valued tolerance relation matrix, as shown in Table A3.

Table A3. Valued tolerance relation matrix, R_A .

R_A	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9	u_{10}	u_{11}	u_{12}	u_{13}	u_{14}	u_{15}	u_{16}	u_{17}	u_{18}	u_{19}	u_{20}
u_1	1.0	0.274	0	0	0	0	0	0	0	0	0	0	0	0	0.034	0	0.034	0.174	0	0
u_2	0.274	1.0	0	0	0	0	0	0	0	0	0	0	0	0	0.174	0.449	0.275	0.449	0	0
u_3	0	0	1.0	0.213	0	0	0	0	0	0	0	0	0	0	0.174	0	0	0	0	0
u_4	0	0	0.213	1.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
u_5	0	0	0	0	1.0	0.174	0	0	0	0	0	0	0	0	0	0	0	0	0	0
u_6	0	0	0	0	0.174	1.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
u_7	0	0	0	0	0	0	1.0	0	0	0	0	0	0	0	0	0	0	0	0	0
u_8	0	0	0	0	0	0	0	1.0	0	0	0.073	0.073	0	0	0	0	0	0	0	0
u_9	0	0	0	0	0	0	0	0	1.0	0.606	0	0	0	0	0	0	0	0	0	0
u_{10}	0	0	0	0	0	0	0	0	0.606	1.0	0	0	0	0	0	0	0	0	0	0
u_{11}	0	0	0	0	0	0	0	0.073	0	0	1.0	0.725	0	0	0	0	0	0	0	0
u_{12}	0	0	0	0	0	0	0	0.073	0	0	0.725	1.0	0	0	0	0	0	0	0	0
u_{13}	0	0	0	0	0	0	0	0	0	0	0	0	1.0	0	0	0	0	0	0	0
u_{14}	0	0	0	0	0	0	0	0	0	0	0	0	0	1.0	0	0	0	0	0	0
u_{15}	0.034	0.174	0.174	0	0	0	0	0	0	0	0	0	0	0	1.0	0.516	0.174	0	0	0
u_{16}	0	0.449	0	0	0	0	0	0	0	0	0	0	0	0	0.516	1.0	0.274	0	0	0
u_{17}	0.034	0.275	0	0	0	0	0	0	0	0	0	0	0	0	0.174	0.274	1.0	0.449	0	0
u_{18}	0.174	0.449	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.449	1.0	0	0
u_{19}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.0	0.449
u_{20}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.449	1.0

Twenty tolerance granules of object $u \in U$ for relation R_A were calculated based on Definition 4, as follows:

$$\begin{array}{ll}
 R_A(u_1) = \{u_1, u_2, u_{15}, u_{17}, u_{18}\}; & R_A(u_2) = \{u_1, u_2, u_{15}, u_{16}, u_{17}, u_{18}\}; \\
 R_A(u_3) = \{u_3, u_4, u_{15}\}; & R_A(u_4) = \{u_3, u_4\}; \\
 R_A(u_5) = \{u_5, u_6\}; & R_A(u_6) = \{u_5, u_6\}; \\
 R_A(u_7) = \{u_7\}; & R_A(u_8) = \{u_8, u_{11}, u_{12}\}; \\
 R_A(u_9) = \{u_9, u_{10}\}; & R_A(u_{10}) = \{u_9, u_{10}\}; \\
 R_A(u_{11}) = \{u_8, u_{11}, u_{12}\}; & R_A(u_{12}) = \{u_8, u_{11}, u_{12}\}; \\
 R_A(u_{13}) = \{u_{13}\}; & R_A(u_{14}) = \{u_{14}\}; \\
 R_A(u_{15}) = \{u_1, u_2, u_3, u_{15}, u_{16}, u_{17}\}; & R_A(u_{16}) = \{u_2, u_{15}, u_{16}, u_{17}\}; \\
 R_A(u_{17}) = \{u_1, u_2, u_{15}, u_{16}, u_{17}, u_{18}\}; & R_A(u_{18}) = \{u_1, u_2, u_{17}, u_{18}\}; \\
 R_A(u_{19}) = \{u_{19}, u_{20}\}; & R_A(u_{20}) = \{u_{19}, u_{20}\}.
 \end{array}$$

From Table A2, we can see that there are ten decision classes: $D_1 = \{u_1, u_2\}$, $D_2 = \{u_3, u_4\}$, $D_3 = \{u_5, u_6\}$, $D_4 = \{u_7, u_8\}$, $D_5 = \{u_9, u_{10}\}$, $D_6 = \{u_{11}, u_{12}\}$, $D_7 = \{u_{13}, u_{14}\}$, $D_8 = \{u_{15}, u_{16}\}$, $D_9 = \{u_{17}, u_{18}\}$, and $D_{10} = \{u_{19}, u_{20}\}$. The lower approximation and upper approximation of each decision class D_l , $l = 1, 2, \dots, 10$, were determined from Definition 5, as follows:

$$\begin{array}{ll}
 (D_1)_A = \phi; & (D_1)^A = \{u_1, u_2, u_{15}, u_{16}, u_{17}, u_{18}\}. \\
 (D_2)_A = \{u_4\}; & (D_2)^A = \{u_3, u_4, u_{15}\}. \\
 (D_3)_A = \{u_5, u_6\}; & (D_3)^A = \{u_5, u_6\}. \\
 (D_4)_A = \{u_7\}; & (D_4)^A = \{u_7, u_8, u_{11}, u_{12}\}. \\
 (D_5)_A = \{u_9, u_{10}\}; & (D_5)^A = \{u_9, u_{10}\}. \\
 (D_6)_A = \phi; & (D_6)^A = \{u_8, u_{11}, u_{12}\}. \\
 (D_7)_A = \{u_{13}, u_{14}\}; & (D_7)^A = \{u_{13}, u_{14}\}. \\
 (D_8)_A = \phi; & (D_8)^A = \{u_1, u_2, u_3, u_{15}, u_{16}, u_{17}\}. \\
 (D_9)_A = \phi; & (D_9)^A = \{u_1, u_2, u_{15}, u_{16}, u_{17}, u_{18}\}. \\
 (D_{10})_A = \{u_{19}, u_{20}\}; & (D_{10})^A = \{u_{19}, u_{20}\}.
 \end{array}$$

The membership degrees of object $u \in U$ belong to $(D_l)_A$ and $(D_l)^A$, $l = 1, 2, \dots, 10$, and were calculated from Definition 6.

$$\begin{aligned}
 \mu_{(D_1)_A}(u_1) &= \min_{v \in R_A(u_1)} (\max(1 - R_A(u_1, v), \hat{\nu})) \\
 &= \min(\max(1 - R_A(u_1, u_1), 1), \max(1 - R_A(u_1, u_2), 1), \\
 &\quad \max(1 - R_A(u_1, u_{15}), 0), \max(1 - R_A(u_1, u_{17}), 0), \max(1 - R_A(u_1, u_{18}), 0)) \\
 &= \min(1.0, 1.0, 0.966, 0.966, 0.826) = 0.826 \\
 \mu_{(D_1)^A}(u_1) &= \max_{v \in R_A(u_1)} (\min(R_A(u_1, v), \hat{\nu})) \\
 &= \max(\min(R_A(u_1, u_1), 1), \min(R_A(u_1, u_2), 1), \min(R_A(u_1, u_{15}), 0), \\
 &\quad \min(R_A(u_1, u_{17}), 0), \min(R_A(u_1, u_{18}), 0)) \\
 &= \max(1.0, 0.274, 0.0, 0.0, 0.0) = 1.0.
 \end{aligned}$$

The results of the membership degrees of each object belong to $(D_l)_A$ and $(D_l)^A$, as shown in Table A4.

To build the fingerprinting decision rules database, we used Definition 7. Moreover, we used Definition 8 to calculate the credibility degree of the rule ρ_1 as follows:

$$\begin{aligned}\mu_A(\rho_1) &= \min_{u \in R_A(u_1)} (\max(1 - R_A(u_1, u), \mu_{D_A}(u))) \\ &= \min(\max(1 - R_A(u_1, u_1), \mu_{(D_1)_A}(u_1)), \max(1 - R_A(u_1, u_2), \mu_{(D_1)_A}(u_2)), \\ &\quad \max(1 - R_A(u_1, u_{15}), \mu_{(D_8)_A}(u_{15})), \max(1 - R_A(u_1, u_{17}), \mu_{(D_9)_A}(u_{17})), \\ &\quad \max(1 - R_A(u_1, u_{18}), \mu_{(D_9)_A}(u_{18}))) \\ &= \min(0.826, 0.726, 0.966, 0.966, 0.826) = 0.726.\end{aligned}$$

Furthermore, we used Definition 9 to determine the support object set of decision rule ρ_1 , as follows:

$$S_A(\rho_1) = R_A(u_1) = \{u_1, u_2, u_{15}, u_{17}, u_{18}\}$$

The fingerprinting decision rules database is shown in Table A5.

Table A5. The fingerprinting decision rules database.

Decision Rule	RSS ₁ (dBm)	RSS ₂ (dBm)	RSS ₃ (dBm)	RSS ₄ (dBm)	RSS ₅ (dBm)	<i>d</i>	Credibility Degree ($\mu_A(\rho_i)$)	Support Object Set ($S_A(\rho_i)$)
ρ_1	-57	-48	-64	-45	-52	RP_1	0.726	$\{u_1, u_2, u_{15}, u_{17}, u_{18}\}$
ρ_2	-58	-49	-63	-48	-53	RP_1	0.551	$\{u_1, u_2, u_{15}, u_{16}, u_{17}, u_{18}\}$
ρ_3	-57	-43	-65	-51	-53	RP_2	0.826	$\{u_3, u_4, u_{15}\}$
ρ_4	-58	-42	-67	-51	-55	RP_2	0.826	$\{u_3, u_4\}$
ρ_5	-55	-54	-63	-52	-57	RP_3	1.0	$\{u_5, u_6\}$
ρ_6	-56	-51	-62	-52	-58	RP_3	1.0	$\{u_5, u_6\}$
ρ_7	-40	-50	-57	-52	-53	RP_4	1.0	$\{u_7\}$
ρ_8	-43	-51	-61	-53	-53	RP_4	0.927	$\{u_8, u_{11}, u_{12}\}$
ρ_9	-41	-53	-62	-39	-50	RP_5	1.0	$\{u_9, u_{10}\}$
ρ_{10}	-42	-53	-63	-39	-49	RP_5	1.0	$\{u_9, u_{10}\}$
ρ_{11}	-50	-49	-61	-53	-50	RP_6	0.927	$\{u_8, u_{11}, u_{12}\}$
ρ_{12}	-50	-50	-61	-53	-51	RP_6	0.927	$\{u_8, u_{11}, u_{12}\}$
ρ_{13}	-50	-56	-65	-45	-48	RP_7	1.0	$\{u_{13}\}$
ρ_{14}	-51	-56	-69	-46	-49	RP_7	1.0	$\{u_{14}\}$
ρ_{15}	-57	-46	-65	-48	-56	RP_8	0.551	$\{u_1, u_2, u_3, u_{15}, u_{16}, u_{17}\}$
ρ_{16}	-58	-47	-64	-50	-55	RP_8	0.551	$\{u_2, u_{15}, u_{16}, u_{17}\}$
ρ_{17}	-59	-49	-63	-47	-56	RP_9	0.551	$\{u_1, u_2, u_{15}, u_{16}, u_{17}, u_{18}\}$
ρ_{18}	-60	-51	-64	-47	-54	RP_9	0.551	$\{u_1, u_2, u_{17}, u_{18}\}$
ρ_{19}	-67	-51	-66	-51	-42	RP_{10}	1.0	$\{u_{19}, u_{20}\}$
ρ_{20}	-66	-53	-66	-51	-43	RP_{10}	1.0	$\{u_{19}, u_{20}\}$

To test the positioning result in the online phase, we collected the RSS values at RP_1 with the set $\theta = (-56, -48, -62, -46, -51)$ (dBm), where each value is from one AP over five APs. Algorithm 2 of the VTRS-DR method was calculated, step by step, as follows.

Step 1: Calculate the valued tolerance relation between θ to the conditions of the rule ρ_i : $R_A(\theta, \rho_i)$, $i = 1, 2, \dots, 20$. The valued tolerance relation $R_A(\theta, \rho_i)$ is greater than zero only for four rules, namely $\rho_1, \rho_2, \rho_{16}$, and ρ_{18} , with $R_A(\theta, \rho_1) = 0.213$, $R_A(\theta, \rho_2) = 0.516$, $R_A(\theta, \rho_{16}) = 0.032$, and $R_A(\theta, \rho_{18}) = 0.174$, respectively.

Step 2: The smallest value $\mu_{\rho_i}(\theta)$ between the valued tolerance relation $R_A(\theta, \rho_i)$ and the credibility degree $\mu_A(\rho_i)$ of rule ρ_i is greater than zero only for four rules, namely $\rho_1, \rho_2, \rho_{16}$, and ρ_{18} , with $\mu_{\rho_1}(\theta) = \min(0.213, 0.726) = 0.213$, $\mu_{\rho_2}(\theta) = \min(0.516, 0.551) = 0.516$, $\mu_{\rho_{16}}(\theta) = \min(0.032, 0.551) = 0.032$, and $\mu_{\rho_{18}}(\theta) = \min(0.174, 0.551) = 0.174$.

Step 3: The credibility degree $\mu_{D_1}(\theta) = \max_{\rho_i \in R(D_1)} (\mu_{\rho_i}(\theta))$ is greater than zero only for three decision classes, namely D_1 , D_8 , and D_9 , with $\mu_{D_1}(\theta) = \max_{\rho_i \in R(D_1)} (\mu_{\rho_i}(\theta)) = \max(\mu_{\rho_1}(\theta), \mu_{\rho_2}(\theta)) = \max(0.213, 0.516) = 0.516$, $\mu_{D_8}(\theta) = 0.032$, and $\mu_{D_9}(\theta) = 0.174$.

Step 4: Because $\exists! D_1 : \mu_{D_1}(\theta) = 0.516$ is the maximum, the new set θ belongs to the decision class D_1 , meaning that Algorithm 2 returns the position of RP_1 (the right classified position); thus, the positioning error is 0 m.

References

- Gabryszuk, J. The Potential to Use Smartphone-based GNSS Receivers for Surveying. *Geomat. Environ. Eng.* **2020**, *14*, 49–57. [\[CrossRef\]](#)
- Nguyen-Huu, K.; Lee, S.-W. A Multi-Floor Indoor Pedestrian Localization Method Using Landmarks Detection for Different Holding Styles. *Mob. Inf. Syst.* **2021**, *2021*, 6617417. [\[CrossRef\]](#)
- Xiao, A.; Chen, R.; Li, D.; Chen, Y.; Wu, D. An Indoor Positioning System Based on Static Objects in Large Indoor Scenes by Using Smartphone Cameras. *Sensors* **2018**, *18*, 2229. [\[CrossRef\]](#)
- Zampella, F.; Jiménez, A.R.; Seco, F. Robust indoor positioning fusing PDR and RF technologies: The RFID and UWB case. In Proceedings of the International Conference on Indoor Positioning and Indoor Navigation, Montbeliard, France, 28–31 October 2013; pp. 1–10.
- Huang, W.; Xiong, Y.; Li, X.; Lin, H.; Mao, X.; Yang, P.; Liu, Y. Shake and walk: Acoustic direction finding and fine-grained indoor localization using smartphones. In Proceedings of the IEEE INFOCOM 2014—IEEE Conference on Computer Communications, 27 April–2 May 2014; pp. 370–378.
- Afzalan, M.; Jazizadeh, F. Indoor Positioning Based on Visible Light Communication: A Performance-based Survey of Real-world Prototypes. *ACM Comput. Surv.* **2019**, *52*, 1–16. [\[CrossRef\]](#)
- Orujov, F.; Maskeliūnas, R.; Damaševičius, R.; Wei, W.; Li, Y. Smartphone based intelligent indoor positioning using fuzzy logic. *Future Gener. Comput. Syst.* **2018**, *89*, 335–348. [\[CrossRef\]](#)
- Al-Madani, B.; Orujov, F.; Maskeliūnas, R.; Damaševičius, R.; Venčkauskas, A. Fuzzy Logic Type-2 Based Wireless Indoor Localization System for Navigation of Visually Impaired People in Buildings. *Sensors* **2019**, *19*, 2114. [\[CrossRef\]](#) [\[PubMed\]](#)
- Yu, J.; Na, Z.; Liu, X.; Deng, Z. WiFi/PDR-integrated indoor localization using unconstrained smartphones. *EURASIP J. Wirel. Commun. Netw.* **2019**, *2019*, 41–59. [\[CrossRef\]](#)
- Ninh, D.B.; He, J.; Trung, V.T.; Huy, D.P. An effective random statistical method for Indoor Positioning System using WiFi fingerprinting. *Future Gener. Comput. Syst.* **2020**, *109*, 238–248. [\[CrossRef\]](#)
- Jiang, P.; Zhang, Y.; Fu, W.; Liu, H.; Su, X. Indoor mobile localization based on Wi-Fi fingerprint's important access point. *Int. J. Distrib. Sens. Netw.* **2015**, *11*, 1–8. [\[CrossRef\]](#)
- Husen, M.N.; Lee, S. Indoor Location Sensing with Invariant Wi-Fi Received Signal Strength Fingerprinting. *Sensors* **2016**, *16*, 1898. [\[CrossRef\]](#)
- Poulose, A.; Kim, J.; Han, D.S. A Sensor Fusion Framework for Indoor Localization Using Smartphone Sensors and Wi-Fi RSSI Measurements. *Appl. Sci.* **2019**, *9*, 4379. [\[CrossRef\]](#)
- Radaelli, L.; Moses, Y.; Jensen, C.S. Using cameras to improve wi-fi based indoor positioning. In Proceedings of the International Symposium on Web and Wireless Geographical Information Systems, Seoul, Korea, 29–30 May 2014; pp. 166–183.
- Shu, Y.; Bo, C.; Shen, G.; Zhao, C.; Li, L.; Zhao, F. Magicol: Indoor Localization Using Pervasive Magnetic Field and Opportunistic WiFi Sensing. *IEEE J. Sel. Areas Commun.* **2015**, *33*, 1443–1457. [\[CrossRef\]](#)
- Wang, W.; Marelli, D.; Fu, M. Fingerprinting-Based Indoor Localization Using Interpolated Preprocessed CSI Phases and Bayesian Tracking. *Sensors* **2020**, *20*, 2854. [\[CrossRef\]](#) [\[PubMed\]](#)
- Yang, C.; Shao, H. WiFi-based indoor positioning. *IEEE Commun. Mag.* **2015**, *53*, 150–157. [\[CrossRef\]](#)
- Gaber, A.; Omar, A. A Study of Wireless Indoor Positioning Based on Joint TDOA and DOA Estimation Using 2-D Matrix Pencil Algorithms and IEEE 802.11ac. *IEEE Trans. Wirel. Commun.* **2015**, *14*, 2440–2454. [\[CrossRef\]](#)
- Chen, H.X.; Hu, B.J.; Zheng, L.L.; Wei, Z.H. An Accurate AoA Estimation Approach for Indoor Localization Using Commodity Wi-Fi Devices. In Proceedings of the 2018 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), Qingdao, China, 14–16 September 2018; pp. 1–5.
- Pawlak, Z. Rough sets. *Int. J. Comput. Inf. Sci.* **1982**, *11*, 341–356. [\[CrossRef\]](#)
- Yang, Y.; Chen, D.; Wang, H. Active Sample Selection Based Incremental Algorithm for Attribute Reduction With Rough Sets. *IEEE Trans. Fuzzy Syst.* **2017**, *25*, 825–838. [\[CrossRef\]](#)
- Tran, A.D.; Arch-int, S.; Arch-int, N. A rough set approach for approximating differential dependencies. *Expert Syst. Appl.* **2018**, *114*, 488–502. [\[CrossRef\]](#)
- Zhao, Y. An algorithm for attribute reduction of incomplete information system. In Proceedings of the 2010 IEEE International Conference on Intelligent Computing and Intelligent Systems, Xiamen, China, 29–31 October 2010; pp. 635–638.
- Renigier-Bižozor, M.; Bižozor, A.; d'Amato, M. Residential market ratings using fuzzy logic decision-making procedures. *Econ. Res. -Ekon. Istraživanja* **2018**, *31*, 1758–1787. [\[CrossRef\]](#)

25. Renigier-Bilozor, M.; Janowski, A.; Walacik, M. Geoscience Methods in Real Estate Market Analyses Subjectivity Decrease. *Geosciences* **2019**, *9*, 130. [[CrossRef](#)]
26. Stefanowski, J.; Tsoukiàs, A. Valued tolerance and decision rules. In Proceedings of the International Conference on Rough Sets and Current Trends in Computing, Banff, AB, Canada, 16–19 October 2000; pp. 212–219.
27. Torres-Sospedra, J.; Montoliu, R.; Trilles, S.; Belmonte, Ó.; Huerta, J. Comprehensive analysis of distance and similarity measures for Wi-Fi fingerprinting indoor positioning systems. *Expert Syst. Appl.* **2015**, *42*, 9263–9278. [[CrossRef](#)]
28. Yang, H.; Zhang, Y.; Huang, Y.; Fu, H.; Wang, Z. WKNN indoor location algorithm based on zone partition by spatial features and restriction of former location. *Pervasive Mob. Comput.* **2019**, *60*, 101085. [[CrossRef](#)]
29. Duong-Bao, N.; He, J.; Thi, L.N.; Nguyen-Huu, K. Analysis of Distance Measures for WiFi-based Indoor Positioning in Different Settings. In Proceedings of the 2022 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET), Meknes, Morocco, 3–4 March 2022; pp. 1–7.
30. Wu, Z.; Jedari, E.; Muscedere, R.; Rashidzadeh, R.J.C.C. Improved particle filter based on WLAN RSSI fingerprinting and smart sensors for indoor localization. *Comput. Commun.* **2016**, *83*, 64–71. [[CrossRef](#)]
31. Babalola, O.P.; Balyan, V. WiFi Fingerprinting Indoor Localization Based on Dynamic Mode Decomposition Feature Selection with Hidden Markov Model. *Sensors* **2021**, *21*, 6778. [[CrossRef](#)] [[PubMed](#)]
32. Zhuang, Y.; Li, Y.; Qi, L.; Lan, H.; Yang, J.; El-Sheimy, N. A Two-Filter Integration of MEMS Sensors and WiFi Fingerprinting for Indoor Positioning. *IEEE Sens. J.* **2016**, *16*, 5125–5126. [[CrossRef](#)]
33. Deng, Z.-A.; Wang, G.; Qin, D.; Na, Z.; Cui, Y.; Chen, J. Continuous Indoor Positioning Fusing WiFi, Smartphone Sensors and Landmarks. *Sensors* **2016**, *16*, 1427. [[CrossRef](#)]
34. Hu, X.; Shang, J.; Gu, F.; Han, Q. Improving Wi-Fi Indoor Positioning via AP Sets Similarity and Semi-Supervised Affinity Propagation Clustering. *Int. J. Distrib. Sens. Netw.* **2015**, *11*, 109642. [[CrossRef](#)]
35. Wang, Y.; Xiu, C.; Zhang, X.; Yang, D. WiFi Indoor Localization with CSI Fingerprinting-Based Random Forest. *Sensors* **2018**, *18*, 2869. [[CrossRef](#)]
36. Chen, Z.; Zou, H.; Yang, J.; Jiang, H.; Xie, L. WiFi Fingerprinting Indoor Localization Using Local Feature-Based Deep LSTM. *IEEE Syst. J.* **2020**, *14*, 3001–3010. [[CrossRef](#)]
37. Song, X.; Fan, X.; He, X.; Xiang, C.; Ye, Q.; Huang, X.; Fang, G.; Chen, L.L.; Qin, J.; Wang, Z. CNNLoc: Deep-Learning Based Indoor Localization with WiFi Fingerprinting. In Proceedings of the 2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation, Leicester, UK, 19–23 August 2019; pp. 589–595.
38. Qin, F.; Zuo, T.; Wang, X. CCpos: WiFi Fingerprint Indoor Positioning System Based on CDAE-CNN. *Sensors* **2021**, *21*, 1114. [[CrossRef](#)]
39. Duong-Bao, N.; He, J.; Vu-Thanh, T.; Thi, L.N.; Do Thi, L.; Nguyen-Huu, K. A Multi-Condition WiFi Fingerprinting Dataset for Indoor Positioning. In Proceedings of the International Conference on Artificial Intelligence and Big Data in Digital Era (ICABDE 2021), Ho Chi Minh, Vietnam, 18–19 December 2021; pp. 1–12.