

Article

# Research on Indoor 3D Positioning Algorithm Based on WiFi Fingerprint

Lixing Wang <sup>1,\*</sup> , Shuang Shang <sup>1</sup> and Zhenning Wu <sup>2</sup><sup>1</sup> School of Computers and Engineering, Northeastern University, Shenyang 110000, China<sup>2</sup> College of Information Science and Engineering, Northeastern University, Shenyang 110000, China

\* Correspondence: wanglixing@mail.neu.edu.cn

**Abstract:** Indoor 3D positioning is useful in multistory buildings, such as shopping malls, libraries, and airports. This study focuses on indoor 3D positioning using wireless access points (AP) in an environment without adding additional hardware facilities in large-scale complex places. The integration of a deep learning algorithm into indoor 3D positioning is studied, and a 3D dynamic positioning model based on temporal fingerprints is proposed. In contrast to the traditional positioning models with a single input, the proposed method uses a sliding time window to build a temporal fingerprint chip as the input of the positioning model to provide abundant information for positioning. Temporal information can be used to distinguish locations with similar fingerprint vectors and to improve the accuracy and robustness of positioning. Moreover, deep learning has been applied for the automatic extraction of spatiotemporal features. A temporal convolutional network (TCN) feature extractor is proposed in this paper, which adopts a causal convolution mechanism, dilated convolution mechanism, and residual connection mechanism and is not limited by the size of the convolution kernel. It is capable of learning hidden information and spatiotemporal relationships from the input features and the extracted spatiotemporal features are connected with a deep neural network (DNN) regressor to fit the complex nonlinear mapping relationship between the features and position coordinates to estimate the 3D position coordinates of the target. Finally, an open-source public dataset was used to verify the performance of the localization algorithm. Experimental results demonstrated the effectiveness of the proposed positioning model and a comparison between the proposed model and existing models proved that the proposed model can provide more accurate three-dimensional position coordinates.



**Citation:** Wang, L.; Shang, S.; Wu, Z. Research on Indoor 3D Positioning Algorithm Based on WiFi Fingerprint. *Sensors* **2023**, *23*, 153. <https://doi.org/10.3390/s23010153>

Academic Editor: Seyed Ali Ghorashi

Received: 15 November 2022

Revised: 13 December 2022

Accepted: 14 December 2022

Published: 23 December 2022



**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/>).

**Keywords:** WiFi fingerprinting; indoor positioning; DNN; TCN

## 1. Introduction

Since the concept of Industry 4.0 was proposed, location-based services (LBS) have attracted significant attention because of their social and commercial value [1] and the technology behind these services has undergone advancement in the field of outdoor positioning. The global navigation satellite system (GNSS), which occupies a dominant position in outdoor positioning [2], has been put into use, bringing great convenience to people's daily lives; however, people generally tend to spend 87–90% of their time indoors [3]. Indoor positioning has broad application prospects, such as guiding customers to shops and food courts in shopping malls, and pushing store information. During an emergency rescue, first responders can use indoor positioning to quickly locate a person in distress, and in terms of personnel and goods management, real-time monitoring of personnel status in chemical plants can be done to avoid factory accidents. In hospitals, indoor positioning is used to provide consultation services and push public service information and in museums, accurate indoor positioning can convert a user's phone into a virtual guide. Although GNSS has been able to more accurately realize outdoor positioning, it is limited by several factors in indoor scene positioning applications. Satellite signals from

the outdoor environment to the indoor environment will suffer from serious attenuation or even complete blockage [4] which in turn causes the devices to receive a very weak signal, and therefore, an inaccurate location is obtained. Indoor positioning remains a problem that needs to be solved, especially in large multistory buildings.

Industries and academia have devoted themselves to research on indoor positioning technology and methods. In the past decade, the number of papers published in the field of indoor positioning has rapidly increased and many wireless technologies have been studied for indoor positioning, including ultrasonic, Bluetooth, radio frequency identification (RFID), ultra-wide bandwidth (UWB), and WiFi technologies. The advantages, disadvantages, and positioning accuracies of these techniques are summarized in Table 1 [5].

**Table 1.** Major indoor positioning technologies.

Positioning Technique	Advantage	Disadvantage	Positioning Accuracy	Popular System
Ultrasonic [6]	High positioning accuracy	Multipath effect, thermal drift effect, high cost, severe decay	Centimeter scale	Bat [7]
RFID [8]	Low cost, high speed	Short distance, low communication capacity	Centimeter scale	LANDMARC [9]
UWB [10]	Strong resistance to interference, strong penetration, high positioning accuracy	High cost	Sub-meter scale	Dart UWB [11]
Bluetooth [12]	Low power consumption, easily deployable	Short distance	Meter scale	BIPS [13]
Infrared [14]	Low-cost, mature technology	Poor resistance to interference	Meter scale	Active Badge [15]
Zigbee [16]	Low-cost, low power consumption	Poor stability	Meter scale	[17]
WiFi [18]	No additional hardware required for deployment, low cost, wide application range	Tedious fingerprint collection, degeneration of WiFi signal	Meter scale	RADAR [19]

Among these technologies, ultrasonic, RFID, UWB, Bluetooth, infrared, and ZigBee indoor positioning techniques require specific hardware support and several positioning beacons, such as infrared, ultrasonic, or RFID beacon whilst WiFi is the most widely deployed indoor wireless infrastructure [20]. The Wi-Fi positioning technique does not require additional hardware and can provide indoor positioning solely using an evenly distributed wireless network set up in the building and the smart device at the user's end. Owing to the increasing popularity of WiFi, WiFi signals are now ubiquitous in society, as they meet the infrastructure requirements of WiFi positioning. In addition, many people have become habituated to using Wi-Fi. Consequently, WiFi positioning has a substantial advantage over other indoor positioning techniques. Therefore, this study focuses on the WiFi fingerprint indoor positioning technology and aims to design an indoor 3D positioning algorithm. First, it constructs temporal fingerprint slices based on sliding windows. The algorithm uses temporal fingerprint slices as inputs. Temporal features are considered and compared with the normal positioning method. In the proposed algorithm, a temporal convolutional network (TCN) feature extractor is used for the automatic extraction of spatiotemporal features to achieve end-to-end indoor 3D positioning.

The remainder of this paper is organized as follows. Section 2 briefly reviews previous research on WiFi fingerprint-based indoor-positioning methods and deep learning. Section 3 describes the framework of the proposed 3D temporal fingerprint algorithm. Section 4 presents a test of the proposed algorithm and presents an analysis of the results. Finally, Section 5 presents conclusions and provides an outlook for future research.

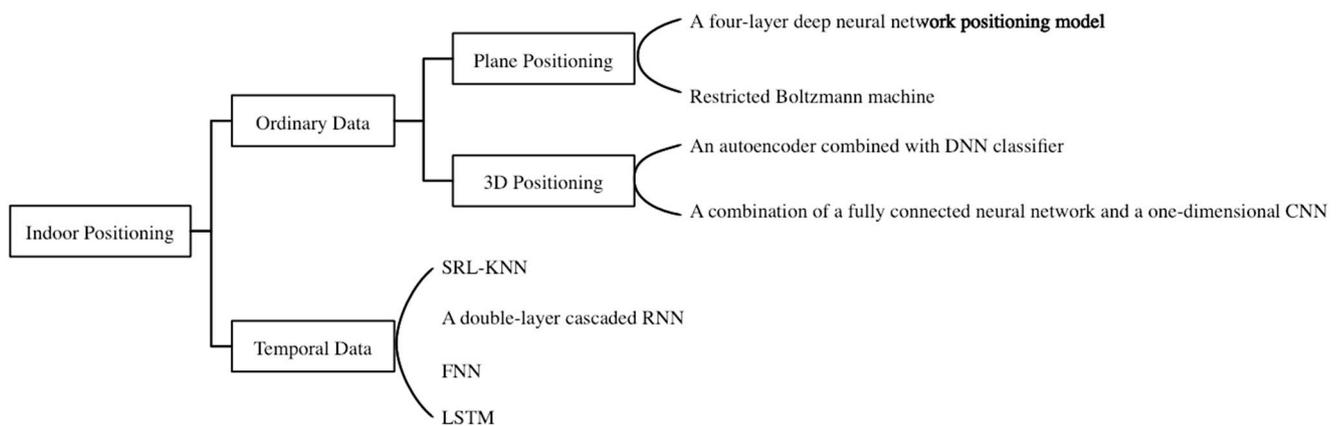
## 2. Literature Review

The fingerprint positioning method must construct a fingerprint map in advance from the target area [21] and a popular method is to construct fingerprint maps in a discrete manner. This approach partitions discrete predefined reference points into an unlocatable environment. The fingerprint is randomly collected at the reference points and did not contain temporal information. Many machine learning algorithms, such as K-nearest neighbor (KNN) [22], naive Bayes [23], support vector machine [24], and random forest [25], are used for plane position estimation. RADAR is the first indoor positioning system to use the location-fingerprint method. This method uses the K-nearest neighbor algorithm, with the core idea being to use the Euclidean distance to measure the similarity of location fingerprints. At present, the improvement of the K-nearest neighbor algorithm in fingerprint-based indoor positioning remains an important research direction. Brunato proposed the weighted K-nearest neighbor (WKNN) positioning algorithm, which assigns different weights to K reference points according to the size of similarity and obtains the location coordinate estimation of the target by calculating the weighted average of coordinates of K neighboring points [26]. Li et al. used Manhattan distance as a similarity measure, which, compared to the Euclidean distance-based WKNN algorithm, improved the accuracy by 33.82% [27]. Tran and Pham proposed a positioning model based on support vector machine (SVM) classification, and found that SVM parameters have a significant impact on positioning performance [24]. Li et al. proposed an indoor positioning prediction model based on an SVM and used an improved sparrow search algorithm to optimize the parameters. The effect of ensemble learning is better than that of a single learner, which is introduced into indoor positioning to further improve positioning accuracy [28]. For example, Lee et al. proposed an indoor positioning model based on random forest classification. With the emergence of deep learning, researchers are attempting to apply deep learning to indoor positioning to achieve better positioning results [29]. In 2016, Zhang et al. introduced deep learning into Wi-Fi fingerprint positioning for the first time. The authors proposed a four-layer deep neural network positioning model that divided the indoor environment into hundreds of square grids, and the number of neurons in the output layer of the neural network classifier was equal to the number of grids [30]. Felix et al. applied deep belief networks and two different types of restricted Boltzmann machines for indoor positioning [31].

For the study of 3D positioning, Sospedra et al. proposed the first open-source multi-floor location fingerprint dataset and adopted the KNN method [32]. Nguyen summarized and compared mainstream WiFi fingerprint-based positioning algorithms such as WKNN, naive Bayes, and neural networks, with experimental results showing that the WKNN algorithm performed well in most cases. In 3D positioning, Clustering is a popular method for floor recognition [33]. Zhou et al. applied the k-means algorithm to cluster a location fingerprint map. The cluster center was used to represent different floors, and the cluster center with the largest similarity to the undetermined site was selected to determine the floor. The selection of the cluster center and K value had a significant influence on the results [34]. Cramariuc et al. proposed an affinity propagation clustering algorithm based on the penalized log-Gaussian distance to perform affinity propagation clustering on fingerprint maps, which improved the performance of fingerprint clustering in multistory buildings. However, the fingerprint data of different layers were still mistakenly assigned to the same cluster [35]. Kim et al. proposed a 3D positioning architecture for an autoencoder combined with a deep neural network (DNN) classifier. The authors used the autoencoder to reduce the dimension of input data, achieving a floor hit ratio of 91.27% and a position coordinate positioning accuracy of 9.29 m [36]. Song et al. proposed different models to predict buildings, floors, and coordinates. Given that buildings are far away from each other, a fully connected neural network was used for building prediction, and a combination of a fully connected neural network and a one-dimensional convolutional neural network was utilized for floor prediction and coordinate prediction, respectively [37].

Another method for constructing fingerprint maps is to use continuous data. This approach involves collecting fingerprints along the trajectory sequentially and recording the location information of the sampling points. The collected fingerprints contained temporal information and the historical information of the walking trajectory can be used to enhance position estimation. Au et al. used a Kalman filter for continuous localization which requires the assumption of Gaussian noise distribution and linear motion [38]. Hoang proposed the SRL (Simultaneous Region Localization)-KNN positioning algorithm, which does not require additional assumptions, and combines the historical location information of the user into the KNN to restrict the search space of the nearest neighbor nodes to the circle. The estimated position at the previous time point was at the center of the circle, the radius is determined by the user's moving speed and the time difference between two consecutive measurements to reduce spatial ambiguity in positioning [39]. Bai et al. applied deep learning to continuous positioning and proposed a positioning model based on a double-layer cascaded RNN. The first RNN layer was used for position matching to match the historical fingerprint measurements with the corresponding estimated position and the second RNN layer is used for position filtering. The output of the first RNN layer is used as the input for the second RNN layer. This method simultaneously trains two RNN networks by minimizing a common loss function to obtain the final accurate position estimate [40]. Khassanov et al. proposed a positioning model that combines a four-layer neural network and long short-term memory (LSTM) to achieve end-to-end continuous indoor positioning [21].

From the research mentioned above, it can be observed that deep learning can be used for indoor positioning to improve positioning accuracy as illustrated in Figure 1. The core of deep learning is a neural network. Neural network algorithms mimic the structure and operation mechanism of brain neural networks, which are connected to a large number of neurons for information transmission [41]. Convolutional neural networks (CNN) contain multilevel nonlinear changes that are classical and widely used structures. The connection between neurons is inspired by the organization of the animal visual cortex [42]. CNNs have developed rapidly over the past decade and have the advantage of maintaining a strong feature extraction ability under a low number of parameters. Google, Facebook, and other business giants prioritized the research on CNN used in temporal modeling until TCN was first proposed by Bai et al. in 2018 [43]. A TCN is a type of special structure that can be used in the sequence of the convolution of the modeling problem of a neural network. It adopts a causal convolution mechanism, an expansion mechanism of convolution, and a residual connection. The TCN was not restricted by the size of the convolution kernel. It can be used to obtain a long history of information. Its structure is more compact, and its parameters can be significantly reduced. The TCN is suitable for solving traditional temporal gradient and gradient explosion problems of the network. Experiments have proven that TCN is superior to RNN and LSTM in terms of the efficiency and accuracy of temporal data analysis. For example, in the field of fault detection, Gao et al. proposed a TCN-based fault diagnosis model for power converters. The model has good robustness and reliability in noisy environments, and it can achieve adaptive feature extraction without any auxiliary pre-denoising algorithm. Compared to the LSTM-based diagnosis model, the fault recognition accuracy was improved by 6%. The number of parameters accounts for 1/4 of the LSTM diagnostic model. This model has advantages in terms of computational efficiency and time [44]. For the load forecasting problem of industrial customers, Wang et al. proposed a short-term load forecasting model for industrial customers based on the TCN and LightGBM (Light Gradient Boosting Machine). They applied a TCN to extract the deep characteristics and temporal relations of the input data. The model effectively reduced redundant features and improved the load-forecasting performance. Compared to the existing LSTM load forecasting model, the proposed model has better robustness and prediction effects [45].

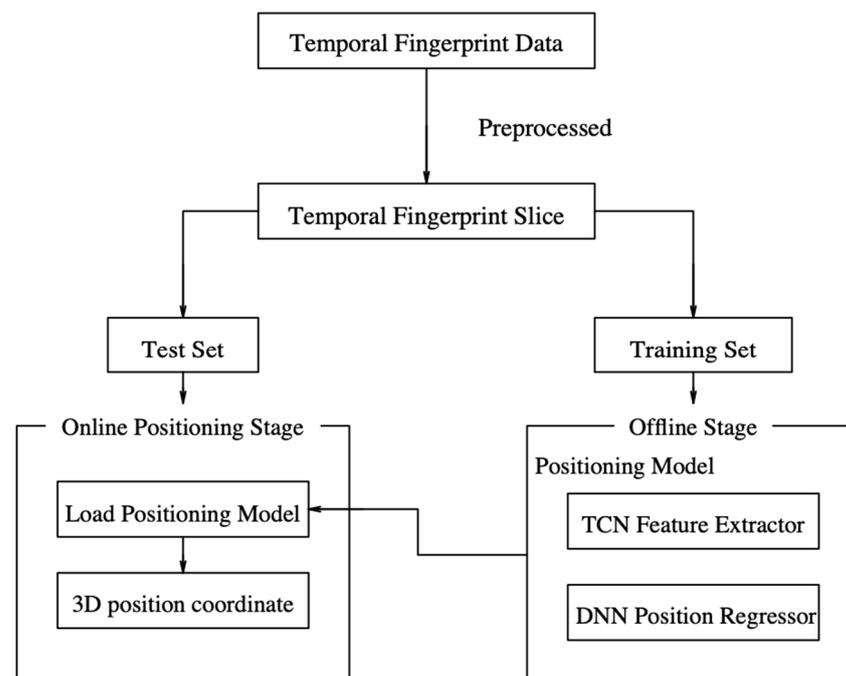


**Figure 1.** Some Deep Learning Methods in WiFi fingerprint-based Indoor Positioning.

In summary, research on indoor positioning has primarily focused on two-dimensional (2D) plane positioning, however, in practical application scenarios, most are multifloor environments, where 2D position coordinates cannot determine the location of the target, and it is necessary to obtain the vertical dimension information, floor, or height of the target in the building. In addition, the traditional location algorithm only considers the position matching of a single fingerprint and ignores the correlation between fingerprint sequences. The current position of the target is related to its historical position. Therefore, it is necessary to develop an indoor 3D positioning model that considers spatiotemporal data. Deep learning, which can be used in 3D positioning model design, has been an effective tool for data analysis in recent years.

### 3. Algorithm Design

This study proposed an indoor 3D positioning algorithm called TCN\_DNNLoc, which is a hybrid algorithm combining TCN and DNN. The proposed algorithm is divided into an offline model training stage and an online fingerprint matching stage. The framework is illustrated in Figure 2.



**Figure 2.** TCN\_DNNLoc framework.

In the offline stage, the temporal fingerprint map is constructed, the 3D positioning model is trained and the fingerprint data are collected in a continuous manner. WiFi fingerprints are collected along the trajectory and the corresponding 3D position coordinates of the collected points are recorded, along with the collected time stamp. Then, the fingerprint map is preprocessed and the RSSI in the fingerprint vector is converted to a number between 0 and 1 for improving the learning ability of the model. Next, a fixed-length sliding time window is used to construct temporal fingerprint slices, which are used as the input of the model, and the TCN feature extractor is used to learn the hidden information and temporal relationship in the input data. Finally, the extracted spatio-temporal features are transmitted to the DNN position regressor for nonlinear mapping of temporal features to 3D coordinates, and the training set is used for model training.

In the online fingerprint-matching stage, the positioning model is first loaded. The collected fingerprint is used to construct the timing fingerprint slice, which is input into the positioning model to realize end-to-end 3D position coordinate prediction. Finally, the positioning result is sent back to the user.

In the following sections, a temporal fingerprint map and its pre-processing method are introduced. Subsequently, a temporal fingerprint slice construction method based on a sliding window is discussed. Next, a fingerprint spatial and temporal feature-learning model based on the TCN is proposed, and finally, the proposed TCN\_DNNLoc 3D temporal fingerprint positioning algorithm is introduced in detail.

### 3.1. Preprocessing of the Temporal Fingerprint Map

#### 3.1.1. Temporal Fingerprint Map

A 3D temporal fingerprint map is constructed continuously. The timestamp, WiFi fingerprint vector, and corresponding 3D position coordinates are recorded by continuous sampling along the trajectory at a certain time interval. The 3D temporal position fingerprint map data are listed in Table 2, with the fingerprints arranged in chronological order.

Table 2. 3D temporal fingerprint map data.

Fingerprint ID	AP <sub>1</sub>	AP <sub>2</sub>	...	AP <sub>n</sub>	x	y	z	Timestamp
1	RSSI <sub>11</sub>	RSSI <sub>12</sub>	...	RSSI <sub>1n</sub>	x <sub>1</sub>	y <sub>1</sub>	z <sub>1</sub>	t <sub>1</sub>
2	RSSI <sub>21</sub>	RSSI <sub>22</sub>	...	RSSI <sub>2n</sub>	x <sub>2</sub>	y <sub>2</sub>	z <sub>2</sub>	t <sub>2</sub>
3	RSSI <sub>31</sub>	RSSI <sub>32</sub>	...	RSSI <sub>3n</sub>	x <sub>3</sub>	y <sub>3</sub>	z <sub>3</sub>	t <sub>3</sub>
4	RSSI <sub>41</sub>	RSSI <sub>42</sub>	...	RSSI <sub>4n</sub>	x <sub>4</sub>	y <sub>4</sub>	z <sub>4</sub>	t <sub>4</sub>
5	RSSI <sub>51</sub>	RSSI <sub>52</sub>	...	RSSI <sub>5n</sub>	x <sub>5</sub>	y <sub>5</sub>	z <sub>5</sub>	t <sub>5</sub>
6	RSSI <sub>61</sub>	RSSI <sub>62</sub>	...	RSSI <sub>6n</sub>	x <sub>6</sub>	y <sub>6</sub>	z <sub>6</sub>	t <sub>6</sub>
....	...	...	...	...	...	...	...	...

The WiFi fingerprint vector collected at time  $t$  can be expressed as shown in (1):

$$FP_t = \{MAC_1 : RSSI_{1t}, MAC_2 : RSSI_{2t}, \dots, MAC_n : RSSI_{nt}\} \quad (1)$$

where  $n$  represents the number of Access Points (Aps) in the environment and  $MAC_i$  represents the MAC address of the  $i$ th AP in an environment. The MAC address is a unique AP identifier and  $RSSI_{it}$  represents the Wi-Fi signal strength value from the  $i$ th AP at time  $t$ . The closer it is to the AP, the greater the RSSI value.

The temporal fingerprint map is defined as shown in (2):

$$\Omega = \{(FP_1, Loc_1), (FP_2, Loc_2), \dots, (FP_m, Loc_m)\} \quad (2)$$

where  $FP_t$  represents the fingerprint vector collected at time  $t$  and  $Loc_t$  is the 3D position vector corresponding to the sampling point at time  $t$ . The 3D position vector is defined as shown in Equation (3):

$$Loc_t = \{x_t, y_t, z_t\} \quad (3)$$

A temporal fingerprint-based indoor positioning problem can be formulated as a sequence learning problem that predicts the user's 3D position vector  $Loc_t$  at time  $t$  based on the current and previous fingerprint vectors, as shown in Equation (4).

$$Loc_t = f(FP_1, FP_2, \dots, FP_t) \quad (4)$$

Here,  $f(\cdot)$  is a sequential function that allows exploiting temporal dependencies in the input.

### 3.1.2. Temporal Fingerprint Map

This section describes the process of preprocessing WiFi RSSI. Through preprocessing, the RSSI value can be converted to a number between zero and one, which is helpful in improving the learning ability of the neural network model. Related studies have proved that WiFi RSSI preprocessing has a significant impact on positioning performance. Nurpeiissov et al. [46] proposed three representation methods for WiFi RSSI preprocessing, namely: zero-one normalized representation, exponential representation, and power representation, as shown in Equations (5)–(7), respectively.

$$ZeroToOneNormalized(RSSI) = \frac{RSSI - c}{-c} \quad (5)$$

$$Exponential(RSSI) = \frac{\exp(\frac{RSSI - c}{\alpha})}{\exp(\frac{-c}{\alpha})} \quad (6)$$

$$Powred(RSSI) = \left(\frac{RSSI - c}{-c}\right)^\beta \quad (7)$$

where the parameter  $c$  is a constant value that should be set to a number less than or equal to the minimum RSSI value to move the RSSI value to a positive range. The zero-one normalization representation performs a linear transformation of the original RSSI and normalizes the RSSI value after moving it to a positive range. The RSSI is more likely to have signal fluctuations when it is far from the AP. Consequently, the exponential representation and power representation perform nonlinear transformations on the RSSI and add penalties for higher RSSI values. The denominator parameter  $\alpha$  is set to a constant and the exponential parameter  $\beta$  is generally set to a mathematical constant  $e$ . After preprocessing, the RSSI value was converted to a value between 0 and 1. An RSSI value close to 1 indicates a stronger signal and 0 indicates an undetected AP.

Preprocessing of the WiFi RSSI has a significant influence on positioning accuracy. For the IMUWIFINE dataset [39] used in this study, the positioning situations of zero-one normalization representation, exponential representation, power representation, and no preprocessing were tested. The RSSI value in the dataset ranges from  $-93$  dBm to  $-34$  dBm and the RSSI value of the undetected AP was set to  $-100$  dBm; thus,  $c$  was set to  $-100$  in this study. The experimental results are shown in Figure 3. It can be observed that WiFi RSSI preprocessing significantly improves the positioning effect. In addition, the power representation has the best effect, and consequently, a power representation is chosen to conduct the RSSI data preprocessing.

### 3.2. Construction of Fingerprint Slice

Traditional positioning algorithms use a single fingerprint as the model input. Owing to the noise and redundancy in WiFi fingerprints, fingerprints collected for the same location may not be the same. Additionally, once the points that are far away from each other in the physical space have a high similarity in the signal space, it will affect the positioning accuracy. A sliding time window to build a temporal fingerprint chip as the input of the positioning model to provide abundant information for positioning. Temporal information can be used to distinguish locations with similar fingerprint vectors and to improve the accuracy and robustness of positioning.

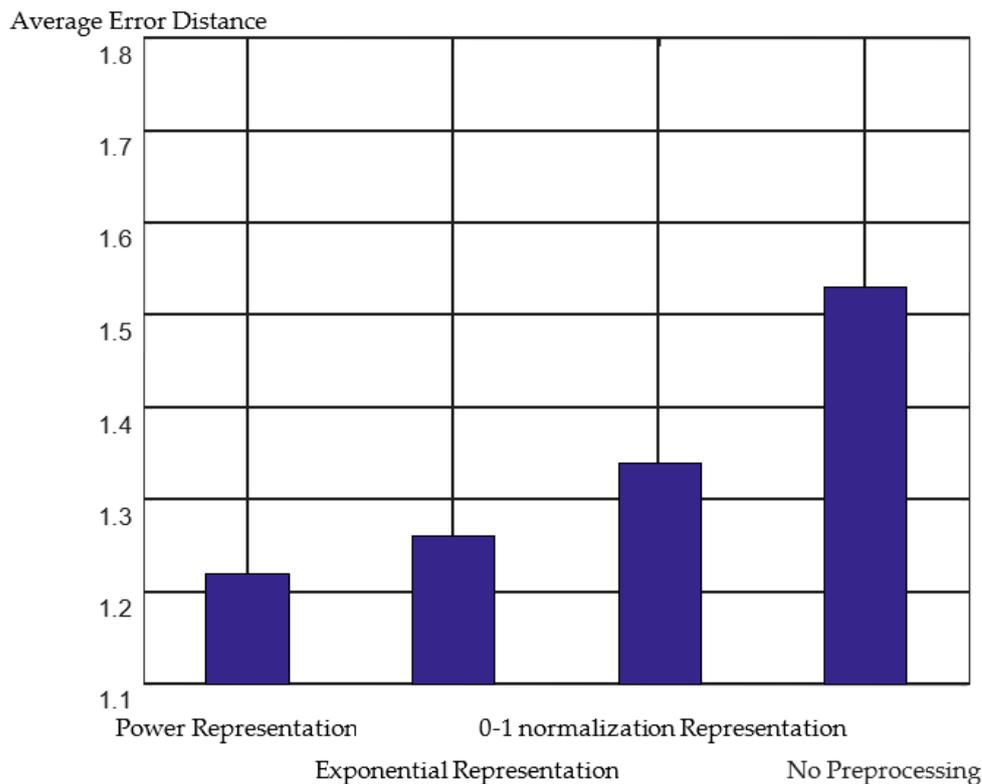


Figure 3. 3D positioning errors of different preprocessing methods.

In this study, a fingerprint vector sequence is selected as the input of the model. Temporal fingerprint slices are constructed by window sliding, where the size of the sliding window was set as  $W$ , and the fingerprint vectors in continuous-time  $t - W + 1$  to time were set  $t$  as a group to form a temporal fingerprint slice. Each temporal fingerprint slice contains  $W$  fingerprint vectors collected continuously. Fingerprint vectors have spatiotemporal dependence and the positioning accuracy can be improved by combining the historical fingerprint information. Considering the size of the window  $W = 4$  as an example, the construction method of the sliding window is introduced, as shown in Figure 4. A window of size four was maintained, and a step of one is taken to continuously slide down on the temporal fingerprint dataset. The fingerprint vector collected from  $t_1$  to  $t_4$  constitutes the fingerprint slice and the fingerprint vector collected from  $t_2$ – $t_5$  constitutes another fingerprint slice. Fingerprinted slices are constructed successively in this manner.

The reconstructed fingerprint slice dataset using the sliding window is shown in Table 3. Each fingerprint slice contains four fingerprint vectors collected continuously, all of which contain time information. The position coordinate corresponding to the last sample in the window is taken as the position label of the entire fingerprint slice.

Table 3. Fingerprint slice dataset.

Fingerprint Slice	Position Coordinate
$[FP_1, FP_2, FP_3, FP_4]$	$x_4, y_4, z_4$
$[FP_2, FP_3, FP_4, FP_5]$	$x_5, y_5, z_5$
$[FP_3, FP_4, FP_5, FP_6]$	$x_6, y_6, z_6$
...	...
$[FP_M, FP_{M+1}, FP_{M+2}, FP_{M+3}]$	$x_{M+3}, y_{M+3}, z_{M+3}$

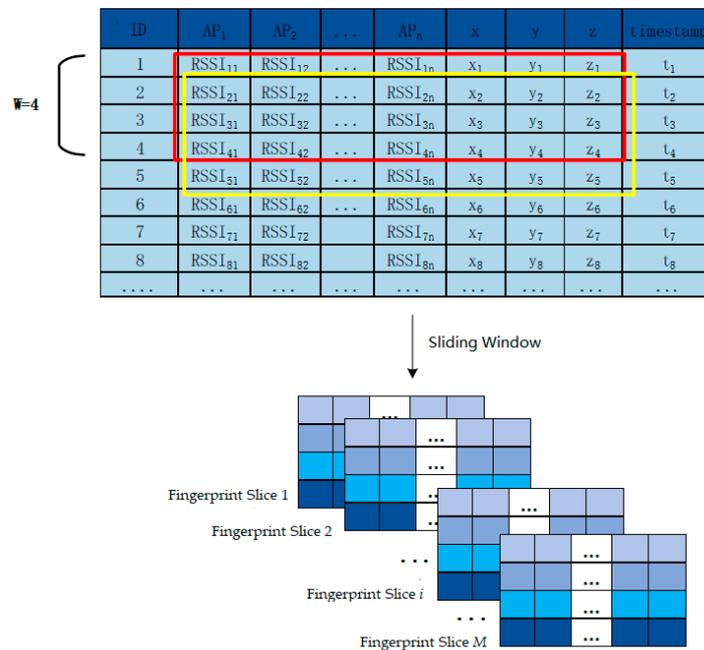


Figure 4. Fingerprint slice construction process.

### 3.3. Positioning Model Design

The model architecture is shown in Figure 5. It is primarily divided into four modules, namely: a fingerprint slice input module, a TCN feature extractor module with three residual blocks, a DNN regressor module with two hidden layers, and a 3D coordinate output module.

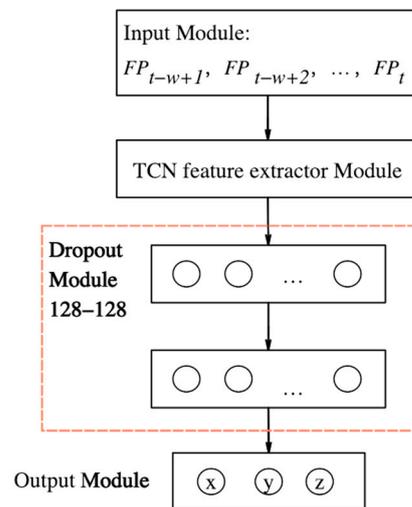


Figure 5. Positioning Model Architecture.

#### 3.3.1. Input Module

The fingerprint vector of a single strip is easily affected by the dynamic changes in the environment and the inherent noise of the signal; therefore, the fingerprint slice  $Series\_FP = [FP_{t-w+1}, FP_{t-w+2}, \dots, FP_t]$  is used as the input of the model.  $Series\_FP$  is a two-dimensional matrix of  $W \times N$ .  $W$  indicates the sequence length of time, which is equal to the size of the sliding window. Here,  $N$  indicates the total number of AP in the environment. The fingerprint slice contains a fingerprint vector collected continuously over time,  $W$ . The

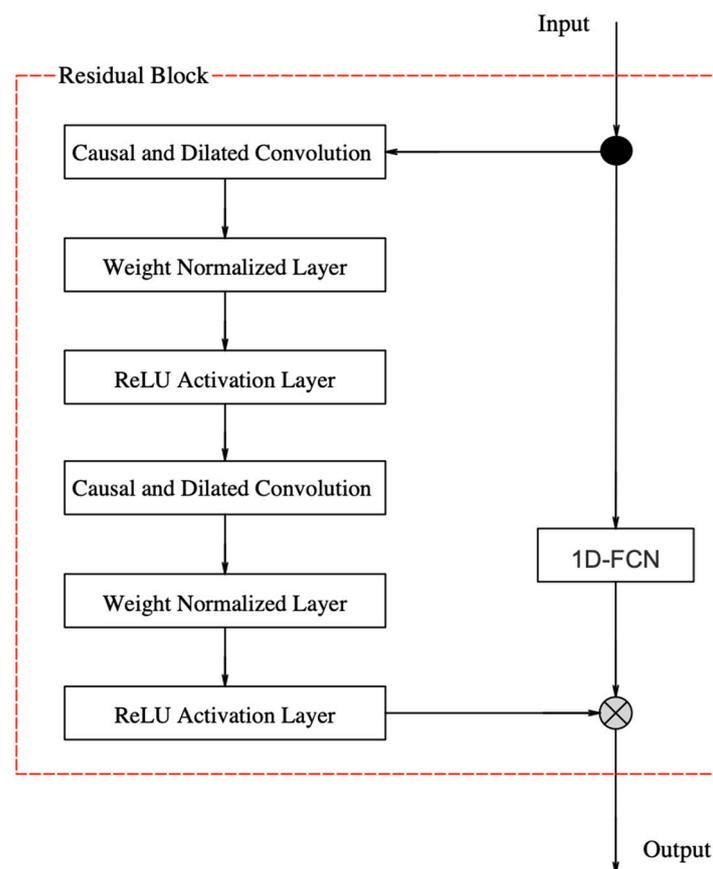
location label corresponding to the fingerprint slice is  $Loc_t = \{x_t, y_t, z_t\}$ , which represents the actual physical location coordinates of the sampling point at time  $t$ .

### 3.3.2. TCN Feature Extractor Module

The TCN feature extractor with three residual blocks is used to extract the features of *Series\_FP*, as shown in Equation (8) and *ST\_Features* indicates the extracted spatio-temporal features.

$$ST\_Features = TCN(Series\_FP) \quad (8)$$

The specific architecture of residual blocks is shown in Figure 6.

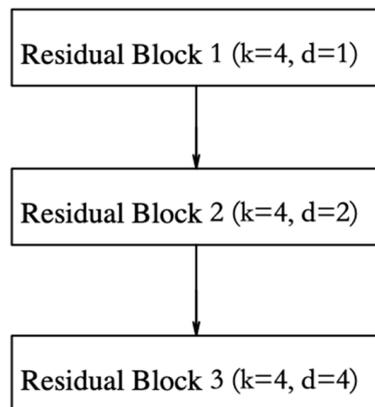


**Figure 6.** TCN residual block architecture.

Each residual block contained two layers, namely: causal and dilated convolutions. Through the causal convolution mechanism, causal constraints are added, and the connections that do not meet the temporal dependence are removed in the training and learning processes. Through the dilated convolution mechanism, larger receptive fields of information are obtained without increasing the number of network layers. The weight-normalized layer is added after each causal and dilated convolution layer to speed up model convergence. The ReLU activation layer activates neurons nonlinearly and adds nonlinear factors to the model to improve its feature learning ability. The Dropout layer sets the loss rate to inactivate neurons proportionally and at random to avoid an excessive fitting model. This is discussed in detail in the next section.

The framework of the TCN feature extractor is shown in Figure 7. where,  $d$  denotes the dilation coefficient. The dilation coefficient of the three residual blocks increases exponentially, which causes the information receptive field to expand by two times, in turn. Here,  $k$  represents the size of the convolution kernel used for feature extraction. The frameworks of the three residual blocks are identical, except for the expansion coefficient.

The residual connection mechanism prevents network degradation, which effectively solves the problems of gradient disappearance and explosion.



**Figure 7.** TCN Feature Extractor.

### 3.3.3. DNN Regressor Module

After extracting spatio-temporal features, the TCN feature extractor is connected to the DNN regressor with two hidden layers to learn the nonlinear mapping between spatio-temporal features and 3D positions, as shown in Figure 6. To prevent network overfitting, a dropout layer is added after each hidden layer. The dropout layer is composed of three neuron outputs for the 3D position coordinate estimation.

The  $ST\_Features$  vector is passed forward to the DNN regressor with two hidden layers, as shown in Equations (9)–(12):

$$y_1 = f(w_1 ST\_Features + b_1) \quad (9)$$

$$y'_1 = drop(y_1) \quad (10)$$

$$y_2 = f(w_2 y'_1 + b_2) \quad (11)$$

$$y'_2 = drop(y_2) \quad (12)$$

Here,  $w_1$  and  $w_2$  are the weights of the two fully connected hidden layers, whilst  $b_1$  and  $b_2$  are the biases of the two hidden layers. The variables  $y_1$  and  $y_2$  are the outputs of the two hidden layers and  $f(\cdot)$  indicates the activation function. ReLU activation function is chosen for the model, which adds nonlinear factors to the model, converts the linear result into a nonlinear output, improves the expression ability of the neural network to the model, and solves the problem that the linear function cannot be fitted. The use of the ReLU activation function and dropout layer can alleviate the overfitting problem to some extent.

### 3.3.4. Output Module

After passing through two fully connected hidden layers, information is transmitted to the output layer with three neurons, which do not use the activation function, as shown in Equation (13):

$$o_i = w_{3i} y'_2 + b_{3i}, i = 1, 2, 3 \quad (13)$$

The  $o_i$  represents the value of each neuron in the output layer,  $o_1$  represents the x-coordinate predicted by the model,  $o_2$  represents the Y-coordinate predicted by the model and  $o_3$  is the Z coordinate predicted by the model. The variables  $w_{3i}$  and  $b_{3i}$  represent the connection weight and bias, respectively, from the second hidden layer to the  $i$ th neuron in the output layer.

The mean square error was used as the loss function of the model, as shown in Equation (14):

$$L_{MSE}(Loc_t, \hat{Loc}_t) = \frac{1}{T} \sum_{t=0}^T \|Loc_t - \hat{Loc}_t\|_2 \quad (14)$$

where  $Loc_t$  denotes the actual 3D coordinates at time  $t$ ,  $\hat{Loc}_t$  denotes the 3D coordinates predicted by the model at time  $t$ . An Adam optimization was used to update the parameters.

## 4. Experiments and Analysis

### 4.1. Dataset

The experiment used the IMUWIFINE dataset [46] launched by ISSAI, which is an open-source position fingerprint dataset with temporal information. The collection site was located in Building C4 on the campus of Nazarbaye University. It covers a total area of over 9000 square meters on three floors: the fourth, fifth, and sixth. The fingerprint vector and corresponding 3D position coordinates were recorded every five seconds along a continuous track. A collection of examples is presented in Figure 8.

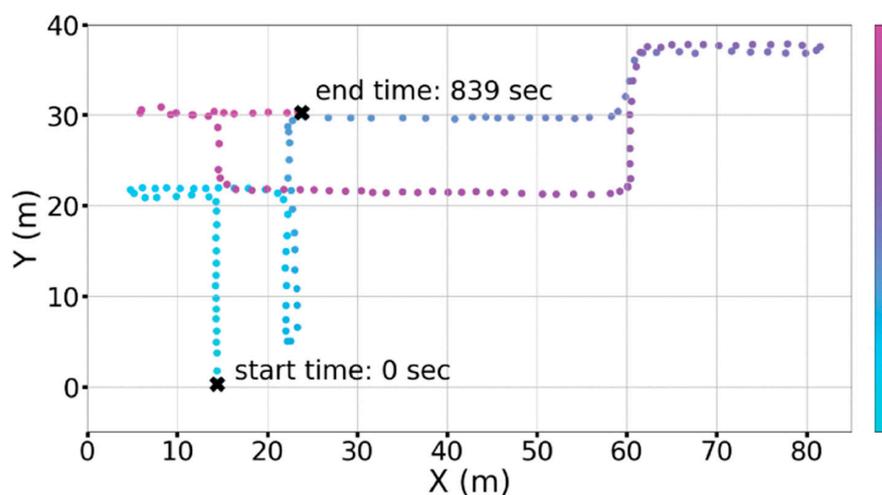


Figure 8. An example of temporal fingerprint collection.

Specific attribute information for the dataset is presented in Table 4. A total of 220 APs were placed in the environment and a default value of  $-100$  dBm was used to represent the undetected APs.

Table 4. IMUWIFINE dataset details.

Attribute	Meaning
AP001-AP220	RSSI from the corresponding AP (dbm)
x	Value on the X-coordinate (m)
y	Value on the Y-coordinate (m)
z	Value on the Z-coordinate (m)
Timestamp	Timestamp of the sample (s)

### 4.2. Evaluation Index

The 3D average error distance and cumulative error distribution functions were used as the performance evaluation indexes of the model.

The definition of 3D average error distance is shown in Equation (15):

$$3D\_MED = \frac{\sum_{i=1}^N dist(Loc_i, \hat{Loc}_i)}{N} \quad (15)$$

where  $Loc_i$  is the actual 3D position coordinate of the target predicted by the positioning model introduced in Section 3.3.  $dist()$  is the Euclidean distance between the actual position  $Loc_i$  and predicted position  $\hat{Loc}_i$ , whilst  $N$  is the sum of the number of samples in all tests.

The  $dist()$  is defined as in Equation (16):

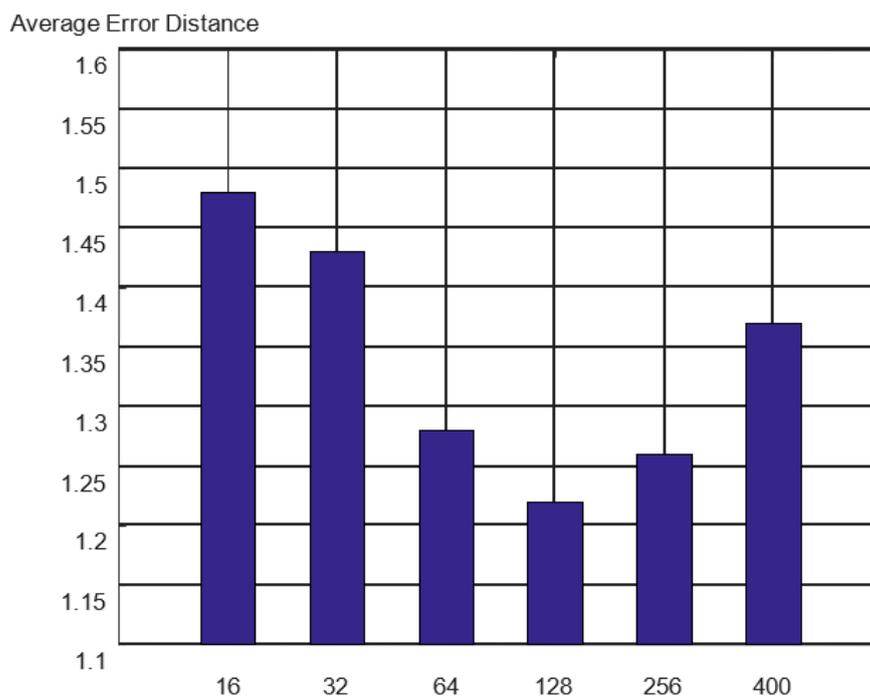
$$dist(Loc, \hat{Loc}) = \sqrt{(x - \hat{x})^2 + (y - \hat{y})^2 + (z - \hat{z})^2} \quad (16)$$

The cumulative error distribution function (CEDF) is another evaluation index and the trend in the error distribution curve indicates the overall error distribution. The faster the curve rises, the better the positioning effect.

#### 4.3. Parameter Setting

To improve the performance of the positioning model, the TCN feature extractor and DNN position regressor were optimized.

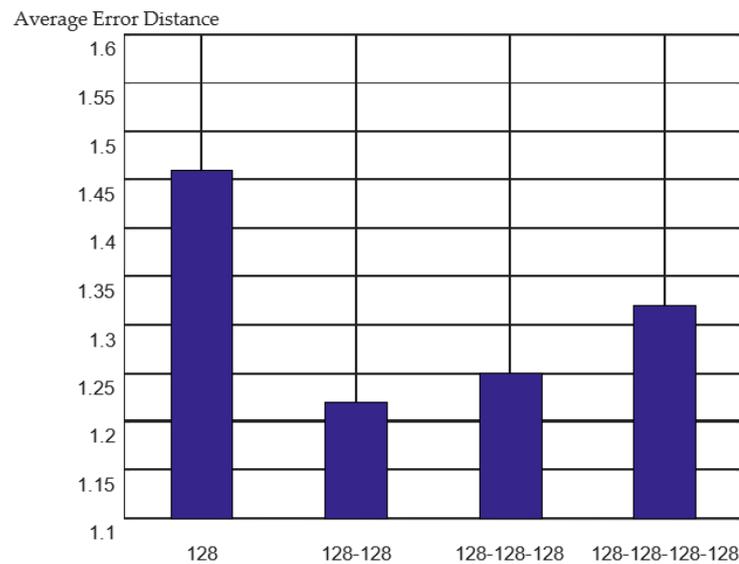
The number of convolution kernels in the TCN feature extractor affects the spatiotemporal feature extraction, which in turn affects the accuracy of the model. This study tested the effect of the TCN with different numbers of convolution kernels. The experimental comparison results are shown in Figure 9.



**Figure 9.** Localization effect of different convolution kernel numbers.

Initially, with an increase in the number of convolution kernels, the positioning precision of the model was greatly improved. The best positioning effect was obtained by using 128 convolution kernels. As the number of convolution kernels continued to increase, the model positioning performance decreased. Additionally, when the number of model parameters increased dramatically, the training time became significantly longer. Consequently, the number of convolution kernels in the TCN space-time feature extractor was set to 128.

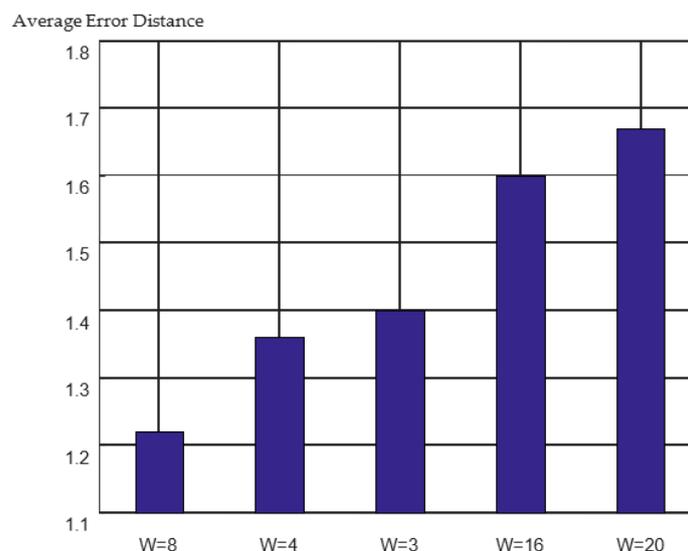
For DNN regressors, different network structures affected the performance of the model. The experimental comparison results are shown in Figure 10.



**Figure 10.** Localization effect of different DNN structures.

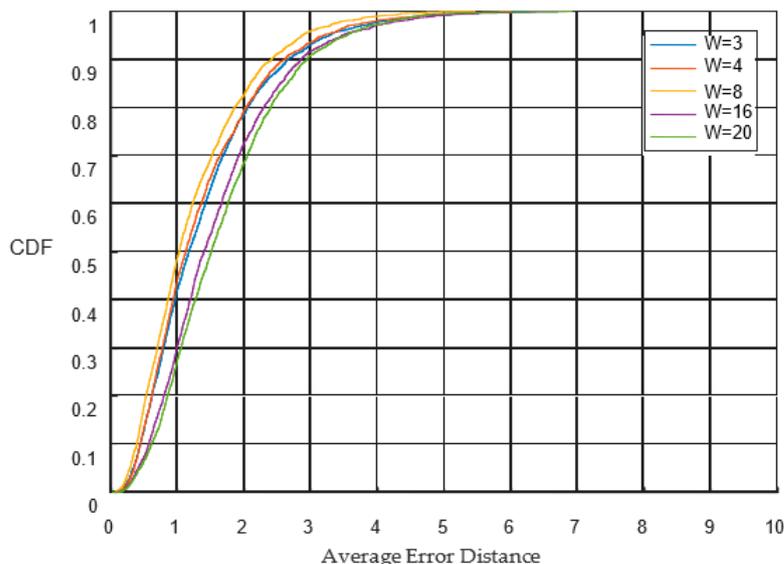
It can be observed that the performance of a single hidden layer was limited. However, increasing the number of network layers can cause overfitting and affect the positioning performance of the model. In the experiment, 128–128 means that two hidden layer structures with 128 neurons were adopted; this selection yielded the best positioning performance.

This study adopted a sliding window to build the temporal fingerprint slice where the length of the temporal fingerprint slice was equal to the size of the sliding window. The longer the series, the more historical information it contains but the more cumulative errors it introduces. To balance these two points for the optimal time series length, experiments were conducted with different series lengths. The positioning errors for the different series lengths are shown in Figure 11.



**Figure 11.** 3D positioning error of different series length.

The minimum average error distance was obtained when the time-series length was eight. The cumulative error distribution curves for different time-series lengths are shown in Figure 12. It can be observed from the figure that when  $W = 8$ , the error distribution curve rises the fastest, and the curve is the shakiest. More than 80% of the positioning results achieved errors of less than two meters. The details can be found in Table 5.



**Figure 12.** Cumulative distribution functions for different sequence lengths.

**Table 5.** The proportion of the average error distance for different time-series lengths.

	<1 m	<2 m	<4 m	<6 m
W = 3	41%	79%	98%	100%
W = 4	45%	79%	98%	100%
<b>W = 8</b>	<b>49%</b>	<b>82%</b>	<b>99%</b>	<b>100%</b>
W = 16	30%	73%	97%	100%
W = 20	28%	69%	97%	100%

In summary, parameter settings of TCN\_DNNLoc algorithm are shown in Table 6.

**Table 6.** TCN\_DNNLoc model parameters.

Parameter	Value
Epochs	1000
Douput	0.1
Residual blocks	3
Convolution kernel size	4
Number of convolution kernels	128
activation function	ReLU
Loss function	Mse
Optimization algorithm	Adam

#### 4.4. Performance Comparison of Different Algorithms

The performance of the proposed TCN\_DNNLoc model was compared with the nonsequential model SRL\_KNN [39], FNN [21], and sequential model LSTM [46]. All hyperparameters were tuned on the validation set, and the final model was evaluated on the test set of IMUWiFine Dataset. For SRL\_KNN, K is set to 3. For FNN, the model is with 5 hidden layers and 128 hidden units. ReLU is used in all hidden layers. More details about the model can be found in Ref. [21]. For LSTM, the model is consisting of four ReLU and three LSTM layers with a dimension size of 128 units. More details about the model can be found in Ref. [46]. The epochs of all the model are set the same as 1000.

The 3D positioning errors of different models are shown in Figure 13.

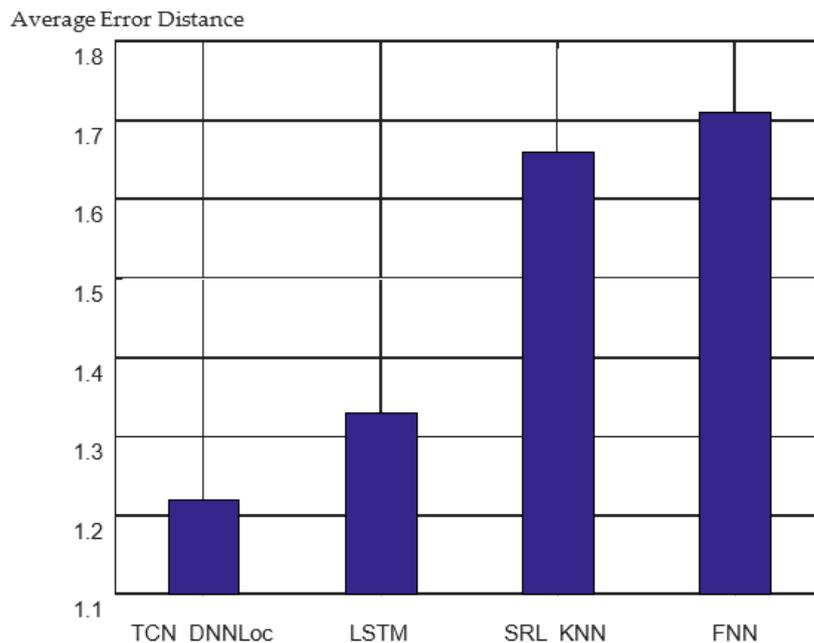


Figure 13. 3D positioning error of different models.

It can be observed from the figure that the positioning accuracy of the sequence model is significantly higher than that of the nonsequence positioning model. For the comparison between the two sequential models, the 3D average error distance of the positioning model based on LSTM was 1.33 m, and the 3D average error distance of the TCN\_DNNLoc was 1.22 m, an accuracy improvement of 7.5%. The cumulative error distribution curves for the different models are shown in Figure 14.

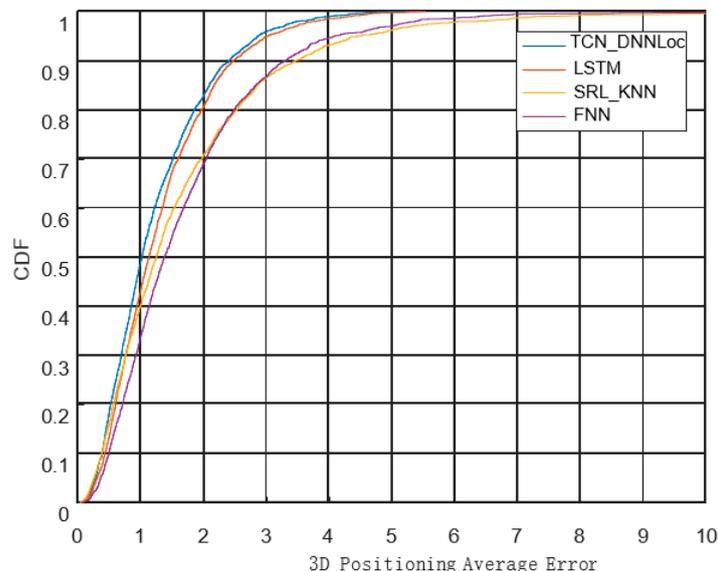


Figure 14. Cumulative distribution functions for different models.

It can be observed from Figure 14 that the error distribution curve of TCN\_DNNLoc increased the fastest, and the curve was the shakiest. More than 80% of the positioning results achieved positioning errors of less than two meters, and the positioning performance was the best, proving the validity of the TCN\_DNNLoc model. The details are included in Table 7.

**Table 7.** The proportion of the average error distance of different models.

	<1 m	<2 m	<4 m	<6 m	Average Error (m)	
					Valid	Test
TCN_DNNLoc	49%	82%	99%	100%	1.21	1.22
LSTM	42%	81%	98%	100%	1.35	1.33
SRL_KNN	40%	70%	93%	96%	1.62	1.66
FNN	34%	69%	94%	97%	1.75	1.71

## 5. Conclusions

In our daily life, 3D indoor positioning is of great significance. It can not only help people navigate in multi-story buildings, but also help managers to count the distribution of people flow in the buildings. At present, it can also help to control the spread of COVID-19 more accurately. This study reviewed relevant references and comprehensively analyzed existing positioning techniques and methods. We focus on the WiFi fingerprint method because of its wide deployment in multi-story buildings. Compared to a single fingerprint input, the fingerprint sequence collected continuously contains historical data, which can provide richer information for positioning. Temporal information can be used to distinguish positions with similar fingerprint vectors and improve the accuracy and robustness of mobile positioning. Based on previous studies, this study applied deep learning techniques to the indoor positioning field and proposed a new sequence-positioning model called TCN\_DNNLoc.

In the proposed model, the sliding time window method is used to construct temporal fingerprint slices, which are used as the input to the model, to provide richer information for positioning, solve the problems of environmental noise and spatial ambiguity, and improve the accuracy and robustness of positioning. A TCN spatiotemporal feature extractor was proposed to extract the sequence fingerprint slice of hidden information and temporal relations. The proposed TCN model adopts causal convolution, dilated convolution, and residual connection mechanisms, and is not restricted by the size of the convolution kernels. The proposed model also expands the receptive with long history information, compact structure, and reduced parameters, and solves the problems of gradient disappearance and gradient explosion. The extracted spatiotemporal features are connected to a deep neural network (DNN) position regressor, and the complex nonlinear mapping relationship between the features and 3D position coordinates is learned to estimate the 3D position coordinates of the target. The effectiveness of the TCN\_DNNLoc model was proved by an experimental comparison with the existing 3D method.

There are two future research directions of this study:

- (1) The positioning accuracy was affected by the connection weight and bias of the model. Future research should consider using a heuristic algorithm to optimize the parameters. Examples include the genetic algorithm (GA) [47], particle swarm optimization (PSO) [48], simulated annealing [49], and quantum annealing algorithms [50]. The optimization algorithms can be used to optimize the connection weight matrix and bias of the positioning model to avoid falling into the local optimal solution. Moreover, the convergence speed of the model can be improved.
- (2) This study considered only WiFi technology for positioning; however, each positioning technology has its own characteristics and limitations. With the development of microelectric systems, the accuracy of sensors has improved continuously, whilst the built-in sensors of mobile phones, such as air-pressure sensors and accelerometers, can be used for auxiliary positioning. Future research must consider the integration of WiFi technology with other technologies to provide more accurate positioning information. Multi-technology fusion and mutual learning can compensate for the limitations of single-positioning technology and improve positioning accuracy.

**Author Contributions:** L.W. conceived the ideas and designed the framework of the algorithm; S.S. implemented the algorithm and conducted the experiments; Z.W. contributed to the RFID system design; L.W. prepared the draft; Z.W. contributed to the manuscript writing and editing. All authors commented on the manuscript. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Natural Science Foundation of China under Grant 71502029 and Grant 61703087.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data presented in this study are openly available in [46].

**Acknowledgments:** The authors would like to acknowledge the financial support from the National Natural Science Foundation of China.

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

## References

- Rajasekar, S.S.; Palanisamy, C.; Saranya, K. Privacy-preserving location-based services for mobile users using directional service fetching algorithm in wireless networks. *J. Ambient Intell. Humaniz. Comput.* **2021**, *12*, 7007–7017. [\[CrossRef\]](#)
- Yassin, A.; Nasser, Y.; Awad, M.; Al-Dubai, A.; Liu, R.; Yuen, C.; Raulefs, R.; Aboutanios, R. Recent advances in indoor localization: A survey on theoretical approaches and applications. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 1327–1346. [\[CrossRef\]](#)
- Chen, R.; Chen, L. Indoor Positioning with Smartphones: The State-of-the-art and the Challenges. *Acta Geod. Cartogr. Sin.* **2017**, *46*, 1316–1326.
- Laoudias, C.; Moreira, A.; Kim, S.; Lee, S.; Wirola, L.; Fischione, C. A survey of enabling technologies for network localization, tracking, and navigation. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 3607–3644. [\[CrossRef\]](#)
- Shang, S.; Wang, L. Overview of WiFi fingerprinting-based indoor positioning. *IEEE Commun.* **2022**, *20*, 3607–3644. [\[CrossRef\]](#)
- Carotenuto, R.; Merenda, M.; Iero, D.; Della Corte, F.G. An indoor ultrasonic system for autonomous 3-D positioning. *IEEE Trans. Instrum. Meas.* **2019**, *68*, 2507–2518. [\[CrossRef\]](#)
- Zhang, W.; Yu, K.; Wang, W.; Li, X. A self-adaptive AP selection algorithm based on multiobjective optimization for indoor WiFi positioning. *IEEE Int. Things J.* **2021**, *8*, 1406–1416. [\[CrossRef\]](#)
- Magnago, V.; Palopoli, L.; Buffi, A.; Tellini, B.; Motroni, A.; Nepa, P.; Macci, D.; Fontanelli, D. Ranging-free UHF-RFID robot positioning through phase measurements of passive tags. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 2408–2418. [\[CrossRef\]](#)
- Monica, S.; Bergenti, F. Hybrid indoor localization using WiFi and UWB technologies. *Electronics* **2019**, *8*, 334. [\[CrossRef\]](#)
- Bernardini, F.; Buffi, A.; Motroni, A.; Nepa, P.; Tellini, B.; Tripicchio, P.; Unetti, M. Particle swarm optimization in SAR-based method enabling real-time 3D positioning of UHF-RFID tags. *IEEE J. Radio Freq. Identif.* **2020**, *4*, 300–313. [\[CrossRef\]](#)
- Xia, J.; Li, S.; Wang, Y.; Jiang, B. Research on UWB/BLE-Based Fusion Indoor Positioning Algorithm and System Application. In Proceedings of the 2021 International Symposium on Computer Technology and Information Science (ISCTIS), Guilin, China, 4–6 June 2021; pp. 50–54.
- Yu, K.; Wen, K.; Li, Y.; Zhang, S.; Zhang, K. A Novel NLOS mitigation algorithm for UWB localization in harsh indoor environments. *IEEE Trans. Veh. Technol.* **2019**, *68*, 686–699. [\[CrossRef\]](#)
- Molina, B.; Olivares, E.; Palau, C.E.; Esteve, M. A multimodal fingerprint-based indoor positioning system for airports. *IEEE Access* **2018**, *6*, 10092–10106. [\[CrossRef\]](#)
- Phutcharoen, K.; Chamchoy, M.; Supanakoon, P. Accuracy Study of Indoor Positioning with Bluetooth Low Energy Beacons. In Proceedings of the 2020 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Computer and Telecommunications Engineering (ECTI DAMT & NCON), Electronics, Pattaya, Thailand, 11–14 March 2020; pp. 24–27.
- Harter, A.; Hopper, A.; Steggle, P.; Ward, A.; Webster, P. The anatomy of a context-aware application. *Wireless Netw.* **2002**, *8*, 187–197. [\[CrossRef\]](#)
- Yelkovan, Y.; Güneren, H.; Akgöz, A.; Eybek, F.; Türk, A.; Göl, İ.; Çetin, Ö. Infrared Beacon Based Sub-Meter Indoor Localization. In Proceedings of the 2014 22nd Signal Processing and Communications Applications Conference (SIU), Trabzon, Turkey, 23–25 April 2014; pp. 1427–1430.
- Li, L.; Ma, J.; Xu, P.; Zhi, T. Summary of study on RFID indoor localisation technology. *Comp. Appl. Soft.* **2015**, *32*, 1–3.
- Bianchi, V.; Ciampolini, P.; De Munari, I. RSSI-based indoor localization and identification for zigbee wireless sensor networks in smart homes. *IEEE Trans. Instrum. Meas.* **2019**, *68*, 566–575. [\[CrossRef\]](#)
- Kuhn, M.J.; Mahfouz, M.R.; Turnmire, J.; Wang, Y.; Fathy, A.E. A Multi-Tag Access Scheme for Indoor UWB Localization Systems Used in Medical Environments. In Proceedings of the 2011 IEEE Topical Conference on Biomedical Wireless Technologies, Networks, and Sensing Systems, Phoenix, AZ, USA, 16–19 January 2011; pp. 75–78.
- Sadowski, S.; Spachos, P. Rssi-based indoor localization with the internet of things. *IEEE Access* **2018**, *6*, 30149–30161. [\[CrossRef\]](#)

21. Khassanov, Y.; Nurpeiissov, M.; Sarkytbayev, A.; Kuzdeuov, A.; Varol, H.A. Finer-Level Sequential Wifi-Based Indoor Localization. In Proceedings of the 2021 IEEE/SICE International Symposium on System Integration (SII), Fukushima, Japan, 11–14 January 2021; pp. 163–169.
22. Yang, Z.; Wu, C.; Liu, Y. Locating in Fingerprint Space: Wireless Indoor Localization with Little Human Intervention. In Proceedings of the 18th Annual International Conference on Mobile Computing and Networking, Istanbul, Turkey, 22–26 August 2021; pp. 269–280.
23. Xiang, P.; Ji, P.; Zhang, D. Enhance RSS-based indoor localization accuracy by leveraging environmental physical features. *Wirel. Commun. Mob. Comput.* **2018**, *2018*, 8956757. [[CrossRef](#)]
24. Tran, D.A.; Pham, C. Fast and Accurate Indoor Localization Based on Spatially Hierarchical Classification. In Proceedings of the 2014 IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems, Philadelphia, PA, USA, 28–30 October 2014; pp. 118–126.
25. Wang, Y.; Xiu, C.; Zhang, X.; Yang, D. WiFi indoor localization with CSI fingerprinting-based random forest. *Sensors* **2018**, *18*, 2869. [[CrossRef](#)]
26. Brunato, M.; Battiti, R. Statistical learning theory for location fingerprinting in wireless LANs. *Comput. Netw.* **2005**, *47*, 825–845. [[CrossRef](#)]
27. Li, C.; Qiu, Z.; Liu, C. An improved weighted k-nearest neighbor algorithm for indoor positioning. *Wirel. Pers. Commun.* **2017**, *96*, 2239–2251. [[CrossRef](#)]
28. Li, H.; Su, J.; Liu, W.; Zhang, Y.; Zhou, X. Indoor Positioning Model Based on Support Vector Regression Optimized by the Sparrow Search Algorithm. In Proceedings of the 2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Krakow, Poland, 22–25 September 2021; pp. 610–615.
29. Lee, S.; Moon, N. Location recognition system using random forest. *J. Ambient Intell. Humaniz. Comput.* **2018**, *9*, 1191–1196. [[CrossRef](#)]
30. Zhang, W.; Liu, K.; Zhang, W.; Zhang, Y.; Gu, J. Deep neural networks for wireless localization in indoor and outdoor environments. *Neurocomputing* **2016**, *194*, 279–287. [[CrossRef](#)]
31. Félix, G.; Siller, M.; Alvarez, E.N. A Fingerprinting Indoor Localization Algorithm Based Deep Learning. In Proceedings of the 2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN), Vienna, Austria, 5–8 July 2016; pp. 1006–1011.
32. Torres-Sospedra, J.; Montoliu, R.; Martínez-Usó, A.; Avariento, J.P.; Arnau, T.J.; Benedito-Bordonau, M.; Huerta, J. UJIIndoorLoc: A New Multi-Building and Multi-Floor Database for WLAN Fingerprint-Based Indoor Localization Problems. In Proceedings of the 2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Busan, Republic of Korea, 27–30 October 2014; pp. 261–270.
33. Nguyen, K.A. A performance guaranteed indoor positioning system using conformal prediction and the WiFi signal strength. *J. Inf. Telecommun.* **2017**, *1*, 41–65. [[CrossRef](#)]
34. Zhou, M.; Yuan, L. Indoor 3D positioning algorithm based on K-means. *Comput. Eng. Des.* **2019**, *40*, 2530–2537.
35. Cramariuc, A.; Huttunen, H.; Lohan, E.S. Clustering Benefits in Mobile-Centric WiFi Positioning in Multi-Floor Buildings. In Proceedings of the 2016 International Conference on Localization and GNSS (ICL-GNSS), Barcelona, Spain, 28–30 June 2014; pp. 1–6.
36. Kim, K.S.; Lee, S.; Huang, K. A scalable deep neural network architecture for multi-building and multi-floor indoor localization based on Wi-Fi fingerprinting. *Big Data Anal.* **2018**, *3*, 4. [[CrossRef](#)]
37. Song, X.; Fan, X.; Xiang, C.; Ye, Q.; Liu, L.; Wang, Z.; He, X.; Yang, N.; Fang, G. A novel convolutional neural network based indoor localization framework with WiFi fingerprinting. *IEEE Access* **2019**, *7*, 110698–110709. [[CrossRef](#)]
38. Au, A.W.S.; Feng, C.; Valaee, S.; Reyes, S.; Sorour, S.; Markowitz, S.N.; Gold, D.; Gordon, K.; Eizenman, M. Indoor tracking and navigation using received signal strength and compressive sensing on a mobile device. *IEEE Trans. Mob. Comput.* **2012**, *12*, 2050–2062. [[CrossRef](#)]
39. Hoang, M.T.; Zhu, Y.; Yuen, B.; Reese, T.; Dong, X.; Lu, T.; Westendorp, R.; Xie, M. A soft range limited K-nearest neighbors algorithm for indoor localization enhancement. *IEEE Sens. J.* **2018**, *18*, 10208–10216. [[CrossRef](#)]
40. Bai, S.; Yan, M.; Wan, Q.; He, L.; Wang, X.; Li, J. DL-RNN: An accurate indoor localization method via double RNNs. *IEEE Sens. J.* **2019**, *20*, 286–295. [[CrossRef](#)]
41. Bishop, C.M. Neural networks and their applications. *Rev. Sci. Instrum.* **1994**, *65*, 1803–1832. [[CrossRef](#)]
42. Gu, J.; Wang, Z.; Kuen, J.; Ma, L.; Shahroudy, A.; Shuai, B.; Liu, T.; Wang, X.; Wang, G.; Cai, J.; et al. Recent advances in convolutional neural networks. *Pattern Recognit.* **2018**, *77*, 354–377. [[CrossRef](#)]
43. Bai, S.; Kolter, J.Z.; Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv* **2018**, arXiv:1803.01271.
44. Yating, G.; Wu, W.; Qiongbina, L.; Fenghuang, C.; Qinqin, C. Fault diagnosis for power converters based on optimized temporal convolutional network. *IEEE Trans. Instrum. Meas.* **2020**, *70*, 3501910. [[CrossRef](#)]
45. Wang, Y.; Chen, J.; Chen, X.; Zeng, X.; Kong, Y.; Sun, S.; Guo, Y.; Liu, Y. Short-term load forecasting for industrial customers based on TCN-LightGBM. *IEEE Trans. Power Syst.* **2020**, *36*, 1984–1997. [[CrossRef](#)]

46. Nurpeiissov, M.; Kuzdeuov, A.; Assylkhanov, A.; Khassanov, Y.; Varol, H.A. End-to-End Sequential Indoor Localization Using Smartphone Inertial Sensors and WiFi. In Proceedings of the 2022 IEEE/SICE International Symposium on System Integration (SII), Narvik, Norway, 9–12 January 2022; pp. 566–571.
47. Whitley, D. A genetic algorithm tutorial. *Stat. Comput.* **1994**, *4*, 65–85. [[CrossRef](#)]
48. Kennedy, J.; Eberhart, R. Particle Swarm Optimization. In Proceedings of the ICNN'95-International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; pp. 1942–1948.
49. Van Laarhoven, P.J.M.; Aarts, E.H.L. Simulated annealing. In *Simulated Annealing: Theory and Applications*; Springer: Dordrecht, The Netherlands, 1987; pp. 7–15.
50. Finnila, A.B.; Gomez, M.A.; Sebenik, C.; Stenson, C.; Doll, J.D. Quantum annealing: A new method for minimizing multidimensional functions. *Chem. Phys. Lett.* **1994**, *219*, 343–348. [[CrossRef](#)]

**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.