

Article

Improved Visible Light-Based Indoor Positioning System Using Machine Learning Classification and Regression

Huy Q. Tran  and Cheolkeun Ha *

Robotics and Mechatronics Lab, University of Ulsan, Ulsan 44610, Korea; quanghuybkdn@gmail.com

* Correspondence: cheolkeun@gmail.com

Received: 11 February 2019; Accepted: 7 March 2019; Published: 13 March 2019



Abstract: Recently, indoor positioning systems have attracted a great deal of research attention, as they have a variety of applications in the fields of science and industry. In this study, we propose an innovative and easily implemented solution for indoor positioning. The solution is based on an indoor visible light positioning system and dual-function machine learning (ML) algorithms. Our solution increases positioning accuracy under the negative effect of multipath reflections and decreases the computational time for ML algorithms. Initially, we perform a noise reduction process to eliminate low-intensity reflective signals and minimize noise. Then, we divide the floor of the room into two separate areas using the ML classification function. This significantly reduces the computational time and partially improves the positioning accuracy of our system. Finally, the regression function of those ML algorithms is applied to predict the location of the optical receiver. By using extensive computer simulations, we have demonstrated that the execution time required by certain dual-function algorithms to determine indoor positioning is decreased after area division and noise reduction have been applied. In the best case, the proposed solution took 78.26% less time and provided a 52.55% improvement in positioning accuracy.

Keywords: indoor positioning system; visible light; machine learning classification; machine learning regression; multipath reflections; signal pre-processing

1. Introduction

In 2007, the Visible Light Communication (VLC) standards CP-1221 (VLC system) and CP-1222 (Visible Light ID system) were established by the Japan Electronics and Information Technology Industries Association (JEITA). CP-1222 is a standard concerned with the field of visible light positioning (VLP) [1]. In addition to the standards that came from Japan, the IEEE Standards Association has published IEEE 802.15.7: Visible Light Communication: Modulation Schemes and Dimming Support [2], and IEEE 802.15.7-2018—IEEE Draft Standard for Local and Metropolitan Area Networks—Part 15.7: Short-Range Optical Wireless Communications [3]. Standardization in VLC may be the catalyst for technological innovation and product commercialization in the near future.

VLC-based indoor positioning provides opportunities to develop highly reliable, robust, and inexpensive positioning technologies [4]. Philips, one of the largest electronics companies in the world, has achieved some initial success in this field. Their solution is to embed Light Emitting Diode (LED) luminaires with VLC technology, and utilize the store lighting to provide location data using the store's mobile app [5]. Additional VLP applications, such as mobile robots, assistive devices for patients with impaired vision or other handicaps are being studied [6–8].

The Global Positioning System (GPS) is the perfect choice for outdoor applications because of its coverage and cost [9]. For indoor positioning, there are several possible options, including

WiFi [10,11], Zigbee-based internet of things (IoT) [12], Bluetooth [13], radio frequency identification (RFID) [14], and camera-based solutions [15]. Depending on many factors, each option achieves a different level of accuracy, but the applied algorithms can be considered a fundamental factor. In this article, we examine some of the indoor positioning studies utilizing machine learning (ML) algorithms, which play a key role in our proposed solution. First, indoor WiFi localization is always the preferred choice, because of the universality of WiFi signals and the availability of many wearable devices with WiFi signal receivers [10]. Akram et al. [11] proposed a novel hybrid indoor WiFi localization that combined soft clustering with the random decision forest algorithm. Zan Li et al. [12] used another ML approach that was applied to the ZigBee-based IoT network. These authors investigated a narrow-band indoor positioning system (IPS) fusing time and received signal strength via ensemble learning. Using a random forest regression model, their solution achieved a 36.1% improvement over the traditional method based on received signal strengths (RSS). For RFID technology, the application of ML to locate an object position in the indoor environment is also efficacious. In Reference [14], to overcome the limitation of the mutual dependence of positioning accuracy and the density of reference tags, an extreme ML algorithm was adopted. The results showed that their solution created a better performance than existing solutions. In addition to using device-based localization systems, there is also the option of using device-free localization (DFL) in a wireless sensor network field to locate a person. In this innovative approach, the application of ML to optimize positioning quality is an inevitable trend. The logistic regression classifier has been used to improve the localization accuracy of a fingerprint-based DFL in a changing environment [16], and an extreme ML algorithm with parameterized geometrical feature extraction for DFL is also suggested in Reference [17].

Several recent studies in the field of VLC-based IPS have applied traditional methods as well as improved ML algorithms for better positioning efficiency [18–21]. Xiansheng Guo et al. [18] proposed an indoor localization solution based on the fusion of multiple classifiers (grid-independent least squares and grid-dependent least squares). This solution produced remarkable results, and obtained 93.03% and 93.15% improvement, respectively, over the RSS ratio and RSS matching methods. Besides reducing positioning errors, the analysis and optimization of other parameters, such as the receiver angle [19] and the LED-ID detection accuracy [20], also contribute significantly to the quality of the system, especially when carried out with the support of ML. However, the influence of reflected waves and noises has not been deeply concerned in these articles. In our previous work [21], the detrimental effects of reflected waves on the VLC-based positioning accuracy as well as incomplete solutions in recent studies have been analyzed in great detail. From this constraint, we proposed a novel method (using kNN-RF) to decrease positioning error in the areas outside the center in the multipath reflection environment without signal pre-processing. We also utilized the importance rate function to reduce computational time, but we found that if we removed too many features in an effort to minimize execution time, the positioning accuracy was reduced.

In this article, to achieve a much higher positioning accuracy and faster computational time under the negative influence of multipath reflection and noises, we suggest an innovative indoor positioning solution. This solution is based on the signal pre-processing technique and dual-function ML algorithms that contain machine learning classification (MLC) and machine learning regression (MLR) functions. The algorithms used were support vector machine (SVM), decision tree (DT), random forest (RF), and k-nearest neighbors (kNN). The obtained results proved that the proposed technique can produce the mean positioning accuracy of 8.75 cm with SVM compared to 19.3 cm with kNN-RF in our previous work [21]. Depending on the selected algorithm, the CPU time also showed a significant improvement with 78.26% in the best case. The encouraging results can be applied to many different indoor environments, from public places (such as supermarkets, theaters, museums, and shopping centers) to private places (such as factories and warehouses). In addition, VLC-based positioning systems can be used for smart home applications and can be used in conjunction with smart canes and smart wheelchairs [8,21]. VLC-based positioning systems can be used to locate the current position of a person carrying an optical sensor, and to help them determine the route to their destination inside a

building. Obviously, applications with VLP using the proposed method are very diverse and useful in the world today.

The main features of our solution can be summarized as follows:

- We first performed noise reduction by using outlier and average filtering. These processes include noise filtering and elimination of low-intensity reflected signals. This type of signal pre-processing improves the accuracy of the later position estimation process.
- Then these data were classified using the classification function of ML algorithms. Data points were assigned to one of two areas: the center area or the edge area. The area division was based on the correlation between the actual location of the receiver on the floor and the RSS from four groups of LED lights suspended from the ceiling. Data division by region is a unique idea that not only significantly reduces total execution time but also contributes to the improvement in positioning accuracy, due to the signal integrity within each individual area.
- After noise reduction and area division, the regression function of the ML algorithms was used to predict the location of the receiver. The results show that the proposed solution greatly improved the execution time and positioning accuracy, despite being influenced by many adverse factors, including noises and reflected waves.
- To evaluate the effectiveness of each ML algorithm, we compared their accuracy in both in the classification process and the regression process after the Cross-Validation (CV) technique was employed to verify the reliability of the algorithm and avoid overfitting. The comparison of positioning accuracy and computational time for all the methods provides a basis for selecting the optimal algorithm for future research.

In the remainder of the article, the proposed system is presented in Section 2. Then, our proposed solution including noise reduction, area division, and location prediction using dual-function ML algorithms are shown in Section 3. In Section 4, we find the optimal parameters for each algorithm. Section 5 offers the simulation performance, some discussions and the comparisons of some popular ML algorithms in terms of computational time and the positioning accuracy. Finally, the conclusion is considered in Section 6.

2. Proposed System

2.1. Simulation Configuration

To analyze the performance of the proposed solution, a typical empty room ($5\text{ m} \times 5\text{ m} \times 2.5\text{ m}$) was assumed, as depicted in Figure 1 [22]. We assumed that there were four LED bulbs suspended from the ceiling at a height of 2.5 m. The total transmitted optical power per LED bulb was fixed at 25 W. The semi-angle of each LED group and the field-of-view (FOV) angle of the photo-detector (PD) were both set at 70° . The four walls were made from plaster that had a reflective rate of 0.7–0.85 [23]. Other important parameters are represented in detail in Table 1.

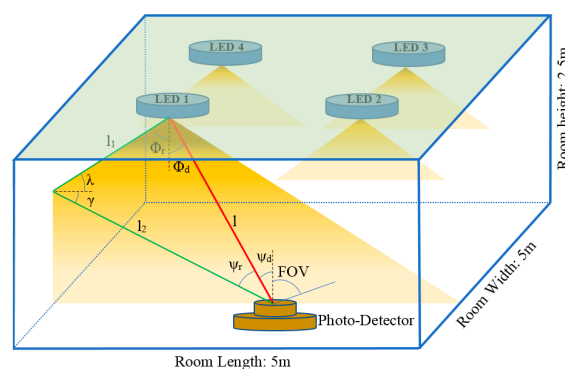
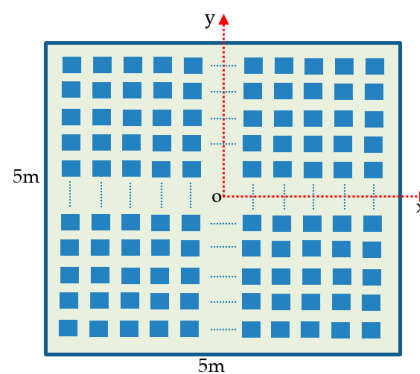


Figure 1. Typical room and channel model.

Table 1. Visible Light Communication (VLC) parameters.

| Object | Parameter | Value |
|---------------------|--|--|
| Simulation space | Room dimension (Length \times Width \times Height) | 5 m \times 5 m \times 2.5 m |
| | Reflective rate | 0.8 |
| Optical transmitter | LED power | 25 W |
| | Number of LED bulbs | 4 |
| | LED bandwidth | 3 MHz |
| | Data rate | 2 Mbps |
| | LED position (x, y, z) (m) | LED 1 (−1.25, −1.25, 2.5) LED 2 (1.25, −1.25, 2.5) LED 3 (1.25, 1.25, 2.5) LED 4 (−1.25, 1.25, 2.5) |
| | Half power semi-angle | 70° |
| Optical receiver | PD active area | 1 cm ² |
| | Field-of-view | 70° |
| | Sensitivity | −30 dBm |
| | Gain of optical filter | 1 |
| | Refractive index of optical concentrator | 1.5 |
| | PD responsivity | 0.54 A/W |

The technique of RSS-based positioning is often used for indoor positioning because of its simplicity [24]. For this work, we develop our solution based on the RSS and fingerprints. We gathered the input data from each reference fingerprint. The distance between adjacent fingerprints was 20 cm, and a 26×26 fingerprint grid was assigned to the floor surface corresponding to 676 reference points. The original coordinates and the fingerprints distribution are shown in Figure 2.

**Figure 2.** Fingerprints on the floor.

2.2. Simulation Configuration VLC Channel and Signal-To-Noise Ratio (SNR) Analysis

To locate the current position of the PD, the intensity of the light source from each LED was modulated, then multiplexing protocol was used to correctly transmit this data from all LED lights to the PD. Finally, demodulating process was executed at the optical receiver [8]. In this paper, an On-Off keying (OOK) modulation using Manchester encoding and Time-division multiplexing (TDM) protocol were used as shown in Figure 3. By using OOK modulation, the logic '1' corresponds to the light 'ON' state, and the logic '0' corresponds to the light 'OFF' state. To send an equal number of positive and negative pulses, Manchester encoding was proposed. After encoding the data, each LED signal was sent to the PD at each time slot by TDM protocol, which divided the overall time into many slots [8].

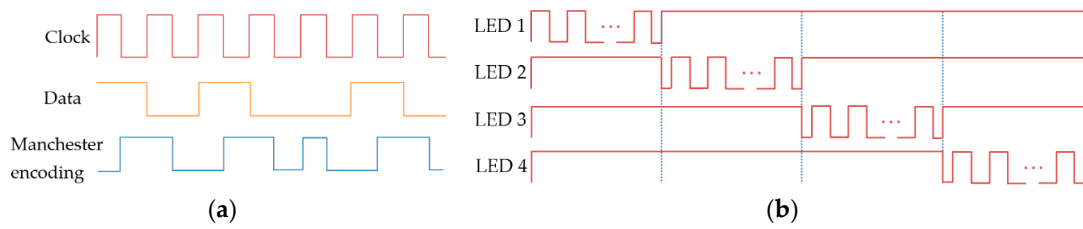


Figure 3. Design of prototype. (a) On-Off keying (OOK) modulation using Manchester encoding. (b) Time-division multiplexing (TDM) protocol.

Depending on the structure of the VLC channel, the RSS and SNR values can be changed. In this work, we analyzed both the line-of-sight (LOS) channel and the diffuse channel. In addition, the influence of multiple noises of the system, including shot noise and thermal noise, was considered in detail. For the direct channel, the DC channel gain is given by the following equation [25]:

$$H_{LOS} = \begin{cases} \left(\frac{n+1}{2\pi}\right)\left(\frac{A}{l^2}\right)\cos^n(\phi)T_s(\Psi_d)g(\Psi_d)\cos(\Psi), & 0 \leq \Psi_d \leq FOV \\ 0, & \Psi_d > FOV \end{cases}, \quad (1)$$

where n is the Lambertian order; A is the active detector area of the PD; l is the distance between the LED and the PD; ϕ is the irradiance angle; Ψ_d is the incidence angle of the directed channel; $T_s(\Psi_d)$ is the gain of the optical filter; $g(\Psi_d)$ is the gain of the optical concentrator; and FOV is the field of view of the PD.

For simplicity, we focus on the effect of the first reflection signal, because other reflected waves have a negligible effect [21]. Thus, the DC channel gain of the first reflection is computed as follows [25]:

$$H_{NLOS} = \begin{cases} \left(\frac{n+1}{2\pi}\right)\left(\frac{A}{l_1^2 l_2^2}\right)\rho\cos^n(\phi)dA_{ref}\cos(\lambda)\cos(\gamma)\cos(\Psi_r)T_s(\Psi_r)g(\Psi_r), & 0 \leq \Psi_r \leq FOV \\ 0, & \Psi_r > FOV \end{cases}, \quad (2)$$

where l_1 is the distance between an LED and a reflective point; l_2 is the distance from a reflective point to the PD; ρ is the reflectance factor; dA_{ref} is the reflective element on the wall; λ is the angle of irradiance from an LED group to the reflective point; γ is the angle between a reflective point and the PD; and Ψ_r is the incidence angle from the wall.

After calculating the DC gain for both direct and diffuse channels, the total received optical power at the PD is as follows [25]:

$$P_{Total} = \sum_1^4 P_T \left(H_{LOS} + \int H_{NLOS} \right), \quad (3)$$

where P_T is the total transmitting optical power.

To compute the SNR, we first determine the Gaussian noise from the output, which is the sum of shot noise σ_S^2 , thermal noise σ_T^2 , and inter-symbol interference σ_{ISI}^2 . However, the σ_{ISI}^2 term can be removed, due to short transmitting duration. The total noise can be expressed as [22]:

$$N_{total}^2 = \sigma_S^2 + \sigma_T^2. \quad (4)$$

Assumed that the dark current noise is small, the shot noise variance due to the received signal and the background radiation is given by [22]:

$$\sigma_S^2 = 2qRP_{Total}B + 2qI_B I_2 B, \quad (5)$$

where q is the electronic charge; R is the PD responsibility; I_B is the photocurrent due to background radiation; I_2 is the noise bandwidth factor; and B is the equivalent noise-bandwidth of the PD.

The thermal noise variance is given by [22]:

$$\sigma_T^2 = \frac{8\pi\kappa T_k}{G} \eta_{PD} A I_2 B^2 + \frac{16\pi^2\kappa T_k \Gamma}{g} \eta_{PD}^2 A^2 I_3 B^3, \tag{6}$$

where κ is the Boltzmann’s constant; T_k is the absolute temperature; G is the open-loop voltage gain; η_{PD} is the fixed capacitance of PD per unit area; Γ is the FET channel noise factor, g is the FET transconductance; and I_3 is the noise-bandwidth factor.

Finally, the SNR for the LOS channel, diffuse channel, and overall channel are depicted in Figure 4, and the SNR equation is as follows [25]:

$$SNR = \frac{R^2 P_{Total}^2}{N_{total}^2}. \tag{7}$$

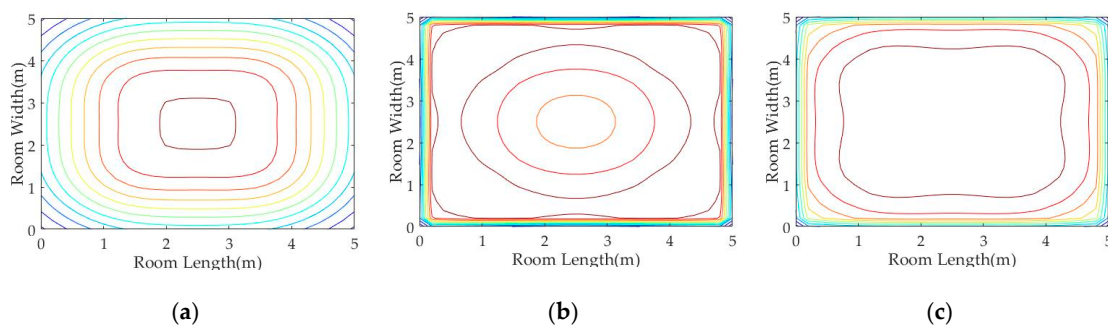


Figure 4. Signal-To-Noise Ratio (SNR) by: (a) line-of-sight (LOS) channel, (b) diffuse channel, and (c) overall channel.

3. Proposed Solution

In an optimal environment, traditional indoor positioning solutions can achieve very high accuracy, particularly when the reflected channel is ignored. However, besides the noises due to sunlight and other electric lights, reflection light always exists, and its intensity may vary according to the current position of the receiver. These reflected signals produce a detrimental effect on positioning accuracy [21]. This is clearly shown in Figure 5 when we use a Trilateration algorithm for indoor positioning composed of two completely opposing environments, one side is the environment that only exists as a directed channel (Figure 5a), and one side is the environment including the highest reflection (Figure 5b). Under the negative impact of multipath reflections, the SNR decreases at the corners and the edges, thus the positioning errors are substantially worse when the PD moves away from the room’s center (see Figure 4) [26]. With the maximum error of approximately 1.5 m occurring at the corners, Figure 5b is a vivid illustration of the negative impact of multipath reflections on positioning accuracy. Therefore, VLC-based indoor positioning techniques must take into consideration all types of noise and reflected signals.

In recent years, ML has made great leaps and has achieved outstanding successes in many fields of science, especially in the field of image processing. However, the application of ML for VLC-based IPS has been quite limited. Challenges when using ML for this application, including sensitivity to noise and high computational times [27], can be considered underlying causes.

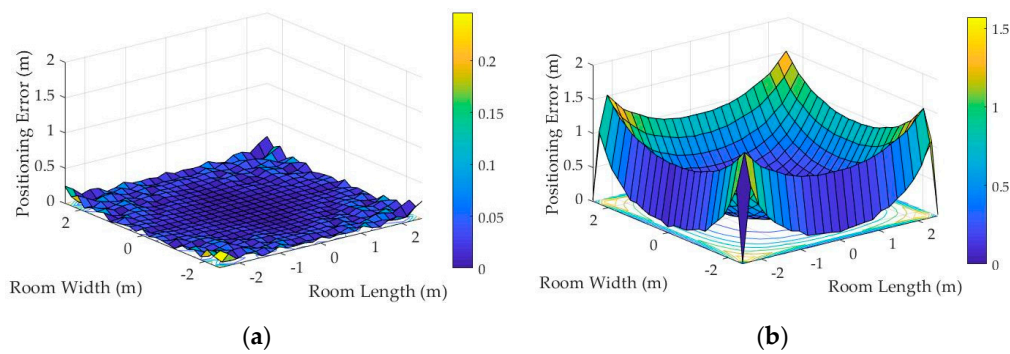


Figure 5. Positioning error using a Trilateration algorithm: (a) without multipath reflections and (b) with multipath reflections.

From the above discussion, we propose an effective solution for VLC-based indoor positioning that improves not only the positioning accuracy but also the execution time (Figure 6). To deal with those two parameters, we combine signal pre-processing techniques and two popular functions of ML-based algorithms (the classification function and the regression function). Signal pre-processing helps filter the noise and eliminates low-intensity reflective signals, thereby reducing the sensitivity to noise. This creates a basis for more accurate positioning later. Then, the two classification and regression functions, in turn, are carried out. The classification process plays a key role in reducing the computational time by dividing the floor surface into two isolated areas, and the regression process helps determine the estimated location of the PD. The whole process is divided into two distinct modes: offline mode and online mode. The main tasks of the offline mode are to collect the RSS from all fingerprints, then in turn conduct noise reduction, area division, and training process. While the online mode gathers online data from the current location of the PD, then the same processes of noise reduction and area division are taken place. However, the difference happens in the final step, when the online mode uses training data in the offline mode to predict the current location of the PD. Further details are discussed in the following Sections.

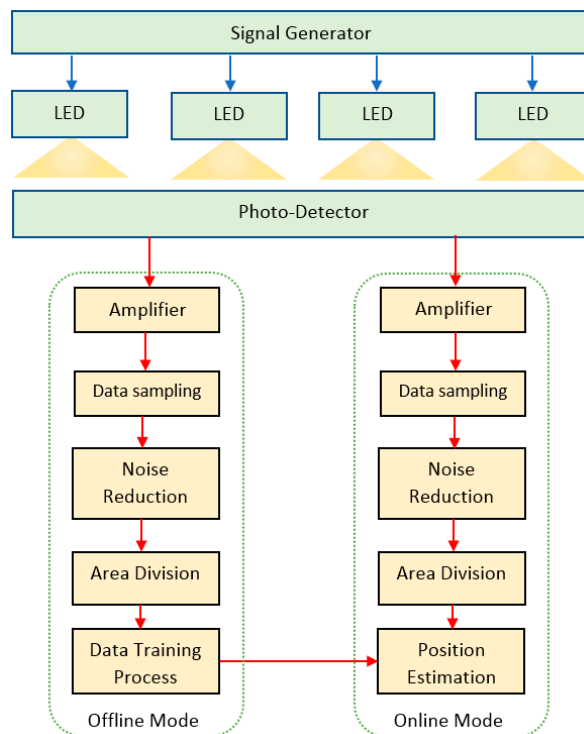


Figure 6. Proposed study method.

3.1. Low-Intensity Reflected Signal Elimination and Noise Reduction

With the LOS channel, the time it takes for the PD to receive the optical signal depends on the distance between the LED light and the PD. With the diffuse channel, it depends on the position of the PD in relation to the reflective point on the wall. At any particular time, the signal received by the PD may be a directed signal from an LED, or a non-directed signal from one of the four walls. The intensity of the reflected signal depends on the position of the receiver as well as the reflective rate of the wall [23]. In Reference [28], to eliminate the diffuse signals, the strongest waves collection was conducted, which can help reduce the impact of multipath reflections. However, in some cases, a combination of a very high reflective rate and other noise sources (i.e., thermal noise and shot noise) can cause the highest signals received at some locations to be diffuse signals. We describe some popular ML algorithms used for area division and location prediction in the next two Sections, while noise sensitivity is a major weakness of ML. It is clear that noise reduction and the elimination of low-intensity reflected signals are important signal pre-processing steps for improving positioning quality. To accomplish this, the following steps were taken:

Step 1: Low Reflected Optical Power Elimination

The received signal from the LED light can be either an LOS signal or a diffuse signal, and this happens randomly according to each sampling. As shown in Figure 7, there is a great difference between the intensity of direct signals and reflected signals at a random point on the floor. If the reflected signal has a very small power compared to direct signals, the training data is no longer uniform. Therefore, eliminating these signal types an important step. To implement this, we calculated the mean value of the RSS based on N sampling data using the following equation:

$$\overline{RSS} = \frac{\sum_{i=1}^N RSS_i}{N}. \quad (8)$$

Next, we used the outlier RSS filter to remove the signals whose power magnitudes are significantly different from the other signals [12]:

$$RSS_i > (1 + \alpha)\overline{RSS}, \quad (9)$$

where α is the outlier ratio and is set to 0.2 in our work after evaluating many different values ($\alpha > 0$).

Step 2: Noise Reduction with Moving Average Filter

After removing the low power reflected signals, we continued to optimize the signals by eliminating the other noise types (i.e., thermal noise and shot noise), which are known as Gaussian noises due to the sunlight from a window or an entrance door [29]. In this Section, we proposed a very simple noise reduction technique called moving average filter [30]. In some cases, this method may not achieve high efficiency if the signal and noise distribution are related to each other [31]. However, in VLC case, the thermal noise and shot noise are signal-independent Gaussian noises [32] and the sum of these random noises is zero in any phase of signal. This filter uses the current and previous $K - 1$ samples to calculate the average RSS:

$$RSS_{new}[n] = \frac{1}{K} \sum_{l=0}^{K-1} RSS[n-l]. \quad (10)$$

After performing this averaging process, the results were utilized as the training dataset for the next steps. The effectiveness of this method is analyzed in detail in Section 5, by comparing cases before and after noise removal.

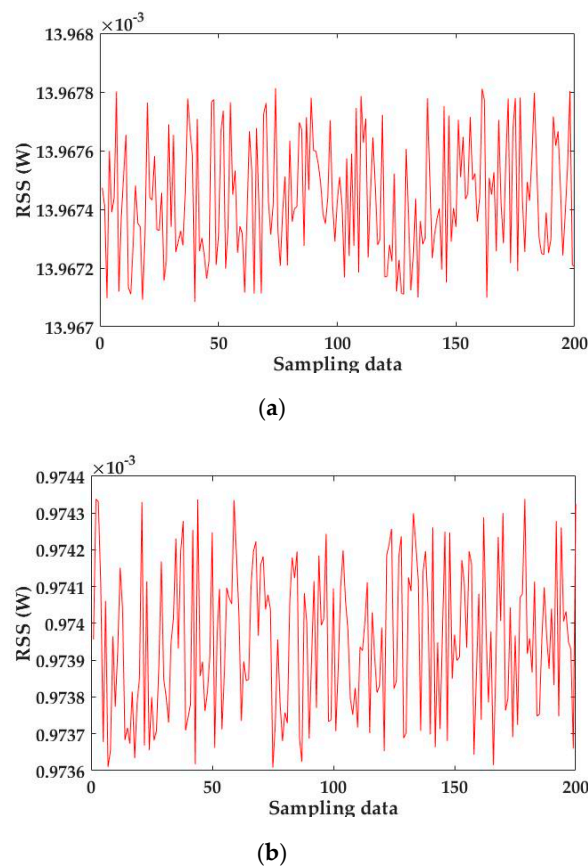


Figure 7. Sampling data by: (a) LOS signals and (b) diffuse signals.

3.2. Area Division with MLC

As discussed in Section 1, computational time is a major constraint when using ML algorithms. In this study, we utilized the classification function of ML to reduce the effect of run time on system performance. This solution is based on the heterogeneous distribution of the SNR (Figure 4). In Figure 4a, the shape of the signal is uniform across the entire floor thanks to the elimination of reflected noise. This homogenous state, however, disappears when the received signal at the PD is a combination of directed and non-directed optical signals (Figure 4c). To conduct area division, we divided the floor of the room into two sections: a center area and an edge area (corners and near-the-wall areas). To prepare for collecting the training dataset, the boundary between the two areas can be determined by two important factors: the identity of the received optical power and the amount of data used for the training process.

Figure 8 shows the average power distribution according to the vertical projection of the room in three reflection level cases: 0.2, 0.5, and 0.8. These values depend directly on the surface material used for the walls, which cause reflective noises [23]. There is a significant difference in the power distribution between the central area (with red spots) and the edge area (with blue spots). It is clear that the central area has a more uniform distribution and is more stable than the edge area, which shows greater reflection intensity.

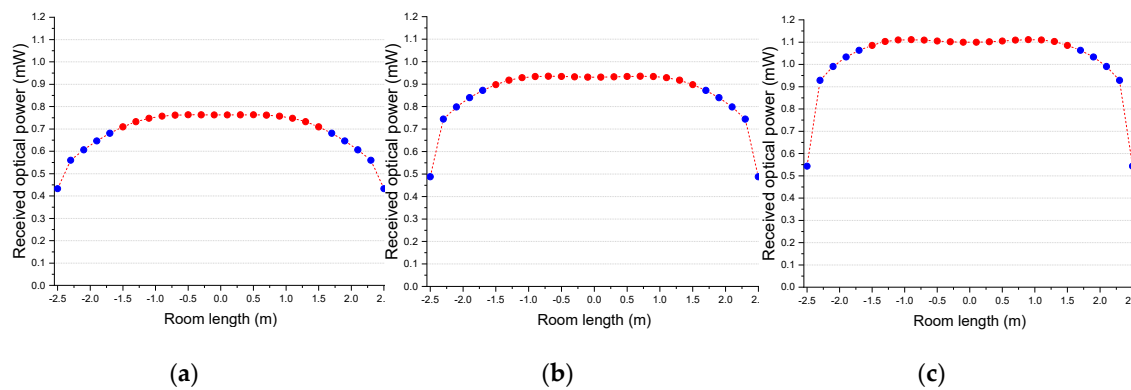


Figure 8. Mean received optical power with different reflection rates by (a) 0.2, (b) 0.5, and (c) 0.8.

To ensure similarity in the amount of training data for the two areas, we set the boundary as shown in Figure 9. By using this division, the training data in the center area and the edge area accounted for 47.93% and 52.07%, respectively, of the total 676 reference points. This balance in the amount of training data helped to reliably assess the effectiveness of the classification process.

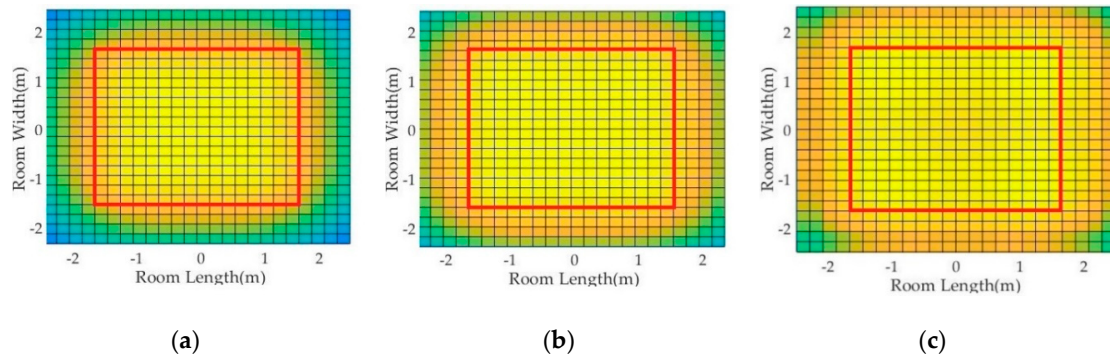


Figure 9. Received optical power with different reflection rates by (a) 0.2, (b) 0.5, and (c) 0.8.

In this study, we used the SVM, DT, kNN, and RF ML algorithms, due to their ability to both classify and regress. After the training and prediction process, we calculated the accuracy score and conducted K-fold CV to evaluate the robustness of each method.

3.3. Location Prediction with MLR

After finishing the noise reduction and area division processes, we estimated the position of the optical receiver as accurately as possible. As mentioned earlier, the ML algorithms used in this article are dual-function, hence SVM, DT, RF, and kNN continue to be adopted with the regression function. From the optimal parameters analysis in the next Section (Table 2), we have the basis to make the final estimation step. This process helps to model a target value based on independent predictors. In this work, the root mean square error (RMSE) was calculated to estimate the skill of the regression predictive model. The results of the location prediction process, as well as the positioning quality comparison of each method, are fully presented and thoroughly explained in Section 5.

Table 2. Optimal parameters.

| Algorithm (Optimized Parameter) | Classification | Regression | | |
|------------------------------------|----------------|-------------|-------------|------------|
| | | Whole Floor | Center Area | Edge Area |
| kNN (k) | 5 | 3 | 3 | 3 |
| SVM (Gamma and C) | 16 and 8 | 128 and 16 | 128 and 16 | 128 and 16 |
| DT (Max-depth) | 8 | 8 | 6 | 8 |
| RF (Tree) | 16 | 12 | 16 | 12 |

4. Tuning Parameters and Results Assessment

Each algorithm has different influential parameters. Optimizing these parameters improves the performance of the algorithms. In this Section, we evaluate algorithm performance by using the CV method and find the optimal parameters for each algorithm [33].

4.1. Algorithms Performance Assessment via K-Fold CV

Two undesirable phenomena that may appear after predicting unseen datasets using ML algorithms are underfitting or overfitting. Underfitting occurs when the model cannot account for the data, and this leads to inaccuracy in the estimated results. Overfitting occurs when there is excellent performance with the available training data, but poor performance with an unknown dataset [34]. To identify these problems, the K-fold CV model evaluation method is a simple and effective approach. In this method, the dataset is divided into K groups and each group is a test set one time, to check and evaluate the performance of the executing algorithm (Figure 10). In this study, K = 10 was used.

$$CV_{K-Fold} = \frac{1}{K} \sum_{j=1}^K MSE_j \tag{11}$$



Figure 10. K-Fold Cross-Validation (CV).

4.2. Parameter Optimization and Accuracy Assessment

4.2.1. The kNN Algorithm

The kNN algorithm is a non-parametric method based on the idea that k nearest samples to the current sample will be determined, then the Euclidean distance is computed. The chosen k value has a profound impact on the performance of the kNN algorithm [21]. Depending on the specific function (classification or regression), the analysis for selecting the k value has certain differences. However, the simplest way to find the k value is to check the average precision score using different k values.

The best overall value of k is chosen, based on the highest precision scores and the highest values of the corresponding CV score. In our case, k was in the range of 2 to 20.

- Classification Function

To optimize the k value for the classification process using the kNN algorithm, the mean of the precision score and CV score was compared for each k value. As depicted in Figure 11, when k increased from 8 to 20, the mean score gradually decreased for both CV and precision. The highest precision score, 0.92, appeared at $k = 5$. At $k = 5$, the CV score was 0.96, which ranks second of all the cases. Therefore, we adopted $k = 5$ as an optimal value for the classification function.

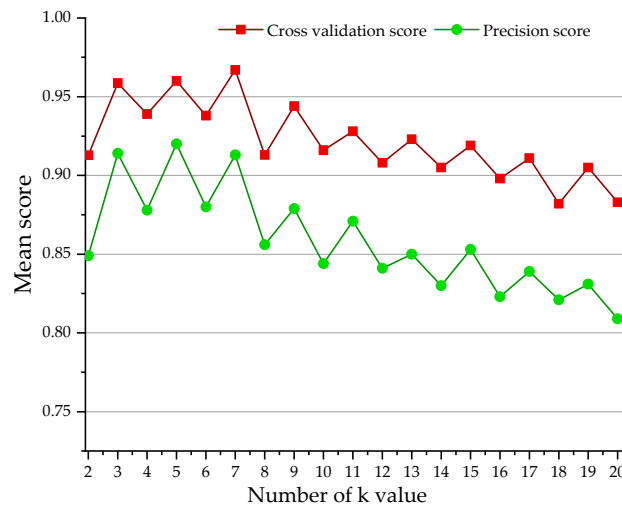


Figure 11. Mean score vs. number of nearest neighbors.

- Regression Function

To maximize the positioning accuracy in the regression process, we conducted a detailed survey of the relationship between the average positioning accuracy, standard deviation, and different k values. In Figure 12a, the best mean of positioning error was obtained when $k = 2$. However, the standard deviation for $k = 2$ was worse than that of $k = 3$, in which the error increased, but not significantly (3.6%). From these considerations, $k = 3$ was used to estimate the position for the whole floor before area division. Similarly, the same values of k were also selected for the center area and the edge area (see Figure 12b,c)

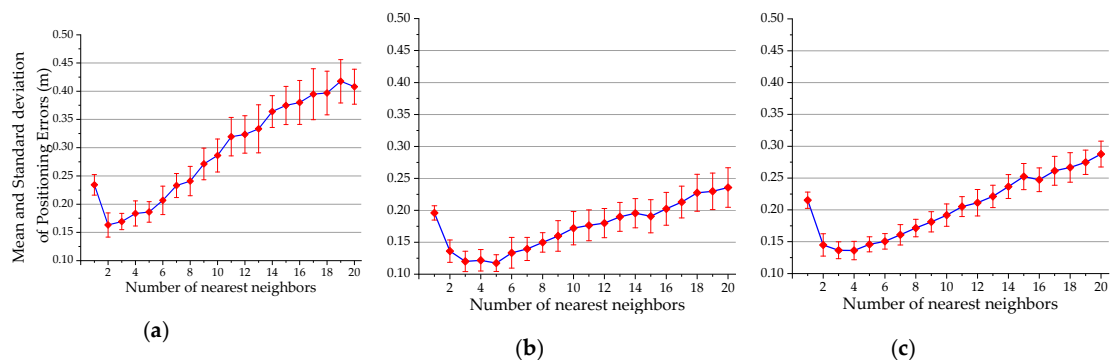


Figure 12. The mean and standard deviation of positioning errors vs. number of nearest neighbors by: (a) whole floor, (b) center area, and (c) edge area.

4.2.2. The SVM Algorithm

SVM is a supervised ML algorithm that can be used to resolve the classification and regression problems, based on a separating hyperplane [34]. In this study, the radial basis function (RBF) kernel was used. Two parameters that need to be optimized for the RBF kernel SVM are Cost and Gamma. When the Cost is small, low bias but high variance may occur, and vice versa. Gamma affects the smoothing of the hyperplane shape [35]. Therefore, optimizing these parameters is a critical procedure. This optimization can be done by the CV method [36]. In this study, a very wide range of Cost values was proposed: from 2^{-5} to 2^4 . Similarly, the Gamma range was from 2^{-2} to 2^7 .

- Classification Function

To evaluate the classification ability of SVM, we used the CV technique to assess how accurately this method can perform in practice. The final mean scores corresponding to the values of Cost and Gamma after 10-fold CV are illustrated in Figure 13. The highest mean score appeared when Cost = 2^3 and Gamma = 2^4 . These values were used throughout this study when the area division process with SVM was employed.

| | | | | | | | | | | | |
|------|----------|--------------|----------|--------|--------|--------|--------|---------------|--------|--------|--------|
| Cost | 2^4 | 0.9143 | 0.9128 | 0.9659 | 0.9823 | 0.9497 | 0.9821 | 0.9941 | 0.9941 | 0.9941 | 0.9941 |
| | 2^3 | 0.9097 | 0.9172 | 0.9571 | 0.9704 | 0.9601 | 0.9631 | 0.9941 | 0.9838 | 0.9896 | 0.9941 |
| | 2^2 | 0.9024 | 0.9113 | 0.9142 | 0.9794 | 0.9838 | 0.9615 | 0.9912 | 0.9867 | 0.9882 | 0.9837 |
| | 2^1 | 0.8002 | 0.9127 | 0.9140 | 0.9216 | 0.9838 | 0.9823 | 0.9585 | 0.9926 | 0.9822 | 0.9926 |
| | 2^0 | 0.6983 | 0.8150 | 0.9113 | 0.9232 | 0.9290 | 0.9822 | 0.9823 | 0.9497 | 0.9926 | 0.9822 |
| | 2^{-1} | 0.7515 | 0.6984 | 0.8256 | 0.9112 | 0.9232 | 0.9290 | 0.9808 | 0.9867 | 0.9497 | 0.9853 |
| | 2^{-2} | 0.5209 | 0.7322 | 0.6985 | 0.8238 | 0.9129 | 0.9229 | 0.9185 | 0.9822 | 0.9792 | 0.9349 |
| | 2^{-3} | 0.5210 | 0.5207 | 0.7411 | 0.6984 | 0.8268 | 0.9128 | 0.9186 | 0.9217 | 0.9822 | 0.9807 |
| | 2^{-4} | 0.5210 | 0.5208 | 0.5207 | 0.7333 | 0.6981 | 0.8299 | 0.9157 | 0.9246 | 0.9186 | 0.9823 |
| | 2^{-5} | 0.5206 | 0.5207 | 0.5209 | 0.5236 | 0.7397 | 0.6982 | 0.8285 | 0.9142 | 0.9230 | 0.9230 |
| | | 2^{-2} | 2^{-1} | 2^0 | 2^1 | 2^2 | 2^3 | 2^4 | 2^5 | 2^6 | 2^7 |
| | | Gamma | | | | | | | | | |

Figure 13. Mean score after CV.

- Regression Function

In a process similar to the classification process above, we first searched for the best positioning accuracy according to Gamma and Cost parameters, and then we verified the robustness of the SVM algorithm corresponding to the values found by 10-fold CV. On the whole floor and edge area, the best positioning accuracy occurred when Cost = 2^4 and Gamma = 2^7 (Figures 14 and 15). There was a minor problem occurring in the remaining case (Figure 16), in that the best tuning parameters were not the choice when reassessed by CV. For instant, the best choice in Figure 16b appeared when Gamma = 2^5 and Cost = 2^4 (orange text), while the best value of Gamma and Cost, in Figure 16a, were 2^7 , and 2^4 , respectively (red text). However, in Figure 16b, the mean CV score in both cases: Gamma = 2^5 , Cost = 2^4 (orange text) and Gamma = 2^7 , Cost = 2^4 (red text) are nearly the same. We, therefore choose Gamma = 2^7 and Cost = 2^4 for our work.

| | | | | | | | | | | | |
|------|-----------------|-----------------|-----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| Cost | 2 ⁴ | 0.3281 | 0.2925 | 0.2607 | 0.2579 | 0.2486 | 0.2347 | 0.1998 | 0.1784 | 0.1592 | 0.1380 |
| | 2 ³ | 0.3817 | 0.2963 | 0.2830 | 0.2692 | 0.2583 | 0.2352 | 0.2489 | 0.2457 | 0.2365 | 0.2048 |
| | 2 ² | 0.5997 | 0.4261 | 0.3098 | 0.2850 | 0.2643 | 0.2577 | 0.2488 | 0.2400 | 0.2382 | 0.2377 |
| | 2 ¹ | 0.7927 | 0.6857 | 0.5335 | 0.3794 | 0.2854 | 0.2646 | 0.2522 | 0.2455 | 0.2449 | 0.2388 |
| | 2 ⁰ | 0.9082 | 0.8316 | 0.7691 | 0.6583 | 0.5119 | 0.3558 | 0.2736 | 0.2672 | 0.2552 | 0.2452 |
| | 2 ⁻¹ | 0.9804 | 0.9147 | 0.8563 | 0.8156 | 0.7305 | 0.6577 | 0.4828 | 0.3393 | 0.2747 | 0.2686 |
| | 2 ⁻² | 1.1033 | 0.9820 | 0.9215 | 0.8823 | 0.8486 | 0.8168 | 0.7428 | 0.6355 | 0.4642 | 0.3402 |
| | 2 ⁻³ | 1.4259 | 1.1167 | 0.9925 | 0.9346 | 0.8793 | 0.8745 | 0.8401 | 0.8050 | 0.7317 | 0.6333 |
| | 2 ⁻⁴ | 1.7403 | 1.4407 | 1.1230 | 0.9733 | 0.9262 | 0.8734 | 0.8728 | 0.8572 | 0.8397 | 0.7995 |
| | 2 ⁻⁵ | 1.8926 | 1.7412 | 1.4389 | 1.1268 | 0.9780 | 0.9412 | 0.8836 | 0.8879 | 0.8748 | 0.8503 |
| | | 2 ⁻² | 2 ⁻¹ | 2 ⁰ | 2 ¹ | 2 ² | 2 ³ | 2 ⁴ | 2 ⁵ | 2 ⁶ | 2 ⁷ |

Gamma
(a)

| | | | | | | | | | | | |
|------|-----------------|-----------------|-----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| Cost | 2 ⁴ | 0.9763 | 0.9818 | 0.9834 | 0.9844 | 0.9859 | 0.9885 | 0.9906 | 0.9928 | 0.9944 | 0.9947 |
| | 2 ³ | 0.9705 | 0.9782 | 0.9822 | 0.9832 | 0.9847 | 0.9853 | 0.9864 | 0.9862 | 0.9890 | 0.9903 |
| | 2 ² | 0.9262 | 0.9631 | 0.9788 | 0.9812 | 0.9839 | 0.9851 | 0.9851 | 0.9856 | 0.9870 | 0.9862 |
| | 2 ¹ | 0.8576 | 0.8978 | 0.9404 | 0.9716 | 0.9806 | 0.9823 | 0.9841 | 0.9852 | 0.9858 | 0.9863 |
| | 2 ⁰ | 0.8140 | 0.8392 | 0.8677 | 0.9042 | 0.9464 | 0.9740 | 0.9819 | 0.9830 | 0.9847 | 0.9857 |
| | 2 ⁻¹ | 0.7837 | 0.8069 | 0.8298 | 0.8446 | 0.8726 | 0.9093 | 0.9511 | 0.9750 | 0.9823 | 0.9835 |
| | 2 ⁻² | 0.7301 | 0.7812 | 0.8041 | 0.8215 | 0.8347 | 0.8497 | 0.8755 | 0.9104 | 0.9531 | 0.9762 |
| | 2 ⁻³ | 0.5663 | 0.7276 | 0.7818 | 0.8027 | 0.8183 | 0.8248 | 0.8363 | 0.8521 | 0.8750 | 0.9122 |
| | 2 ⁻⁴ | 0.3132 | 0.5639 | 0.7314 | 0.7786 | 0.8007 | 0.8182 | 0.8246 | 0.8316 | 0.8383 | 0.8529 |
| | 2 ⁻⁵ | 0.1676 | 0.3284 | 0.5658 | 0.7327 | 0.7793 | 0.8034 | 0.8164 | 0.8187 | 0.8240 | 0.8334 |
| | | 2 ⁻² | 2 ⁻¹ | 2 ⁰ | 2 ¹ | 2 ² | 2 ³ | 2 ⁴ | 2 ⁵ | 2 ⁶ | 2 ⁷ |

Gamma
(b)

Figure 14. The mean score in the whole area of: (a) positioning accuracy and (b) CV.

| | | | | | | | | | | | |
|------|-----------------|-----------------|-----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| Cost | 2 ⁴ | 0.5005 | 0.3805 | 0.2932 | 0.2392 | 0.2011 | 0.1630 | 0.1400 | 0.1204 | 0.1043 | 0.0960 |
| | 2 ³ | 0.6367 | 0.5658 | 0.4565 | 0.3435 | 0.2628 | 0.2262 | 0.1973 | 0.1711 | 0.1447 | 0.1236 |
| | 2 ² | 0.7511 | 0.6738 | 0.6250 | 0.5538 | 0.4478 | 0.3204 | 0.2562 | 0.2058 | 0.1940 | 0.1781 |
| | 2 ¹ | 0.9406 | 0.7679 | 0.6871 | 0.6484 | 0.5979 | 0.5471 | 0.4370 | 0.3065 | 0.2422 | 0.2121 |
| | 2 ⁰ | 1.4013 | 0.9354 | 0.7653 | 0.6869 | 0.6813 | 0.6353 | 0.6139 | 0.5529 | 0.4346 | 0.3019 |
| | 2 ⁻¹ | 1.9358 | 1.3835 | 0.9364 | 0.7809 | 0.7065 | 0.6809 | 0.6631 | 0.6486 | 0.6136 | 0.5532 |
| | 2 ⁻² | 2.2525 | 1.9334 | 1.3773 | 0.9335 | 0.7707 | 0.7063 | 0.6885 | 0.6926 | 0.6663 | 0.6514 |
| | 2 ⁻³ | 2.4090 | 2.2307 | 1.9144 | 1.3680 | 0.9355 | 0.7841 | 0.7239 | 0.6931 | 0.6757 | 0.6723 |
| | 2 ⁻⁴ | 2.4725 | 2.3979 | 2.2300 | 1.9215 | 1.3774 | 0.9306 | 0.7589 | 0.7261 | 0.6827 | 0.6925 |
| | 2 ⁻⁵ | 2.5251 | 2.4762 | 2.3914 | 2.2333 | 1.9317 | 1.3647 | 0.9542 | 0.7696 | 0.7006 | 0.6975 |
| | | 2 ⁻² | 2 ⁻¹ | 2 ⁰ | 2 ¹ | 2 ² | 2 ³ | 2 ⁴ | 2 ⁵ | 2 ⁶ | 2 ⁷ |

Gamma
(a)

| | | | | | | | | | | | |
|------|-----------------|-----------------|-----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| Cost | 2 ⁴ | 0.9635 | 0.9791 | 0.9873 | 0.9916 | 0.9937 | 0.9960 | 0.9972 | 0.9979 | 0.9984 | 0.9986 |
| | 2 ³ | 0.9389 | 0.9522 | 0.9683 | 0.9836 | 0.9892 | 0.9928 | 0.9940 | 0.9958 | 0.9968 | 0.9977 |
| | 2 ² | 0.9068 | 0.9300 | 0.9419 | 0.9549 | 0.9701 | 0.9858 | 0.9905 | 0.9929 | 0.9941 | 0.9951 |
| | 2 ¹ | 0.8705 | 0.9089 | 0.9267 | 0.9313 | 0.9418 | 0.9554 | 0.9719 | 0.9867 | 0.9913 | 0.9929 |
| | 2 ⁰ | 0.7284 | 0.8744 | 0.9074 | 0.9223 | 0.9294 | 0.9352 | 0.9401 | 0.9551 | 0.9728 | 0.9867 |
| | 2 ⁻¹ | 0.4585 | 0.7425 | 0.8735 | 0.9086 | 0.9216 | 0.9266 | 0.9281 | 0.9327 | 0.9393 | 0.9548 |
| | 2 ⁻² | 0.2375 | 0.4550 | 0.7453 | 0.8707 | 0.9060 | 0.9211 | 0.9260 | 0.9254 | 0.9288 | 0.9346 |
| | 2 ⁻³ | 0.1146 | 0.2293 | 0.4572 | 0.7489 | 0.8743 | 0.9081 | 0.9208 | 0.9250 | 0.9250 | 0.9237 |
| | 2 ⁻⁴ | 0.0384 | 0.0978 | 0.2223 | 0.4585 | 0.7461 | 0.8774 | 0.9060 | 0.9176 | 0.9232 | 0.9251 |
| | 2 ⁻⁵ | 0.0020 | 0.0349 | 0.0893 | 0.2352 | 0.4549 | 0.7466 | 0.8756 | 0.9060 | 0.9209 | 0.9245 |
| | | 2 ⁻² | 2 ⁻¹ | 2 ⁰ | 2 ¹ | 2 ² | 2 ³ | 2 ⁴ | 2 ⁵ | 2 ⁶ | 2 ⁷ |

Gamma
(b)

Figure 15. The mean score in the edge area of: (a) positioning accuracy and (b) CV.

| | | | | | | | | | | | |
|------|-----------------|-----------------|-----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| Cost | 2 ⁴ | 0.2747 | 0.1998 | 0.1458 | 0.1236 | 0.1029 | 0.0889 | 0.0799 | 0.0775 | 0.0784 | 0.0775 |
| | 2 ³ | 0.2886 | 0.2620 | 0.2095 | 0.1637 | 0.1281 | 0.1143 | 0.0959 | 0.0842 | 0.0783 | 0.0776 |
| | 2 ² | 0.3035 | 0.2782 | 0.2552 | 0.2283 | 0.1897 | 0.1439 | 0.1212 | 0.1066 | 0.0936 | 0.0851 |
| | 2 ¹ | 0.3356 | 0.2858 | 0.2787 | 0.2550 | 0.2387 | 0.2091 | 0.1757 | 0.1372 | 0.1199 | 0.1058 |
| | 2 ⁰ | 0.5145 | 0.3158 | 0.2854 | 0.2699 | 0.2621 | 0.2474 | 0.2367 | 0.2094 | 0.1713 | 0.1351 |
| | 2 ⁻¹ | 0.8388 | 0.4953 | 0.3151 | 0.2845 | 0.2696 | 0.2566 | 0.2553 | 0.2421 | 0.2280 | 0.2041 |
| | 2 ⁻² | 1.1270 | 0.8307 | 0.5006 | 0.3080 | 0.2777 | 0.2634 | 0.2580 | 0.2534 | 0.2474 | 0.2412 |
| | 2 ⁻³ | 1.2859 | 1.1357 | 0.8361 | 0.4937 | 0.3142 | 0.2830 | 0.2674 | 0.2540 | 0.2548 | 0.2469 |
| | 2 ⁻⁴ | 1.3650 | 1.2827 | 1.1349 | 0.8243 | 0.4881 | 0.3070 | 0.2788 | 0.2679 | 0.2492 | 0.2584 |
| | 2 ⁻⁵ | 1.4057 | 1.3678 | 1.2775 | 1.1186 | 0.8371 | 0.4924 | 0.3149 | 0.2850 | 0.2704 | 0.2560 |
| | | 2 ⁻² | 2 ⁻¹ | 2 ⁰ | 2 ¹ | 2 ² | 2 ³ | 2 ⁴ | 2 ⁵ | 2 ⁶ | 2 ⁷ |

Gamma
(a)

| | | | | | | | | | | | |
|------|-----------------|-----------------|-----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| Cost | 2 ⁴ | 0.9663 | 0.9822 | 0.9910 | 0.9931 | 0.9950 | 0.9963 | 0.9970 | 0.9971 | 0.9971 | 0.9970 |
| | 2 ³ | 0.9598 | 0.9684 | 0.9797 | 0.9880 | 0.9921 | 0.9938 | 0.9956 | 0.9967 | 0.9970 | 0.9970 |
| | 2 ² | 0.9570 | 0.9625 | 0.9676 | 0.9753 | 0.9845 | 0.9906 | 0.9926 | 0.9943 | 0.9959 | 0.9966 |
| | 2 ¹ | 0.9508 | 0.9590 | 0.9639 | 0.9678 | 0.9727 | 0.9782 | 0.9862 | 0.9911 | 0.9930 | 0.9946 |
| | 2 ⁰ | 0.8934 | 0.9550 | 0.9596 | 0.9650 | 0.9682 | 0.9707 | 0.9739 | 0.9798 | 0.9868 | 0.9914 |
| | 2 ⁻¹ | 0.7093 | 0.8946 | 0.9563 | 0.9606 | 0.9647 | 0.9680 | 0.9698 | 0.9712 | 0.9752 | 0.9773 |
| | 2 ⁻² | 0.3950 | 0.7055 | 0.8962 | 0.9559 | 0.9622 | 0.9649 | 0.9671 | 0.9685 | 0.9700 | 0.9717 |
| | 2 ⁻³ | 0.2143 | 0.3945 | 0.6974 | 0.8987 | 0.9567 | 0.9614 | 0.9655 | 0.9681 | 0.9688 | 0.9688 |
| | 2 ⁻⁴ | 0.0920 | 0.1822 | 0.3865 | 0.7168 | 0.8983 | 0.9576 | 0.9620 | 0.9669 | 0.9679 | 0.9691 |
| | 2 ⁻⁵ | 0.0257 | 0.1023 | 0.2155 | 0.3961 | 0.7130 | 0.8991 | 0.9570 | 0.9615 | 0.9655 | 0.9676 |
| | | 2 ⁻² | 2 ⁻¹ | 2 ⁰ | 2 ¹ | 2 ² | 2 ³ | 2 ⁴ | 2 ⁵ | 2 ⁶ | 2 ⁷ |

Gamma
(b)

Figure 16. The mean score in the center area of: (a) positioning accuracy and (b) CV.

4.2.3. The DT Algorithm

DT is one of the most useful and simple ML algorithms for classification and regression [36]. However, irrelevant attributes within the training data can result in overfitting. This problem can be resolved using the CV method [37]. To optimize the accuracy of DT, the maximum depth of the tree (max-depth) needs to be attained. In the next step, we analyzed the optimal values for the two cases of classification and regression. The max-depth values were manually chosen using the precision score and CV score shown in Figures 17 and 18.

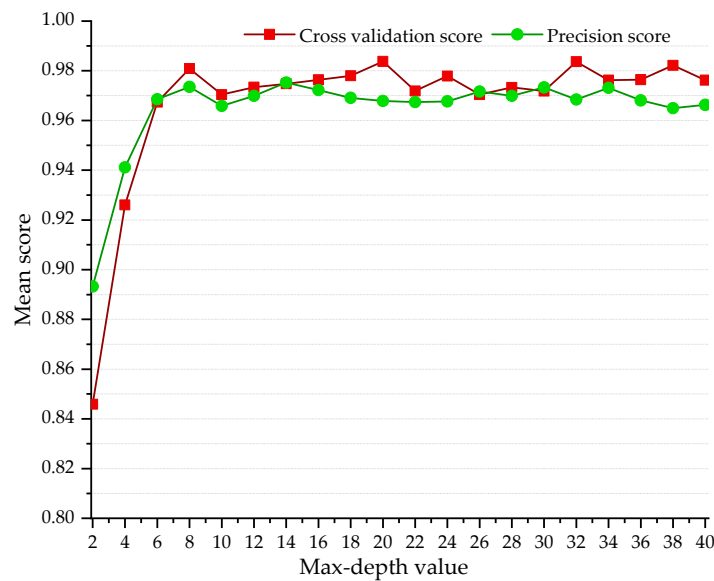


Figure 17. Mean score vs. max-depth.

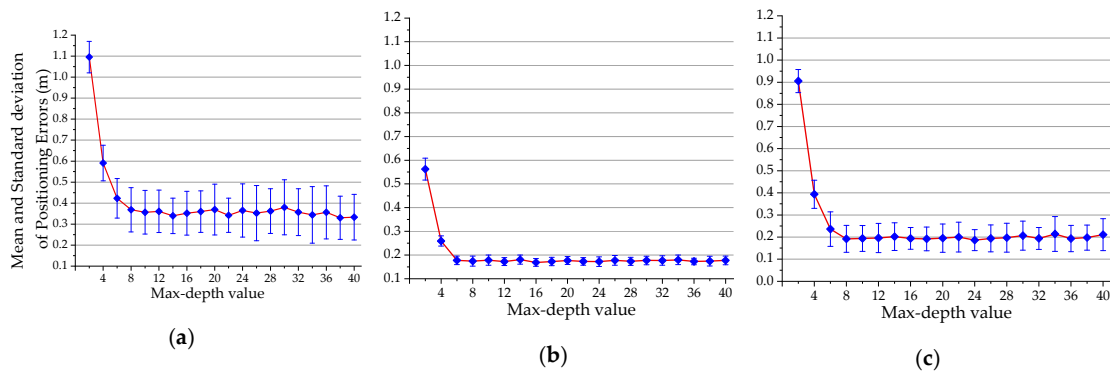


Figure 18. Mean and standard deviation of positioning errors vs. max-depth by: (a) whole floor, (b) center area, and (c) edge area.

• Classification Function

As can be seen in Figure 17, the mean precision score and CV score gradually improve as the max-depth value increases. However, both these scores are nearly stable when max-depth ≥ 8 . To avoid time-consuming computer processing, the smallest possible value is chosen. In this case, max-depth = 8 was the best choice.

• Regression Function

In Figure 18a, the mean and standard deviation of the positioning error were inversely proportional to the max-depth of the tree when the max-depth was in the range of 2 to 8. Then the errors remained almost unchanged as we continued to increase the max-depth. As mentioned earlier, higher values of max-depth result in longer execution times. We therefore chose max-depth = 8 for the whole floor. Similarly, we chose max-depth = 6 for the center area (Figure 18b), and max-depth = 8 for the edge area (Figure 18c).

4.2.4. The RF Algorithm

Like the DT algorithm, the RF algorithm is capable of operating under classification and regression functions. RF is made of a bunch of decision trees using a random subset of data [34]. The number of trees is the most important parameter to optimize with RF algorithms [38]. Based on the data shown

in Figures 19 and 20, the selected trees for both cases (i.e., classification and regression) are shown in Table 2.

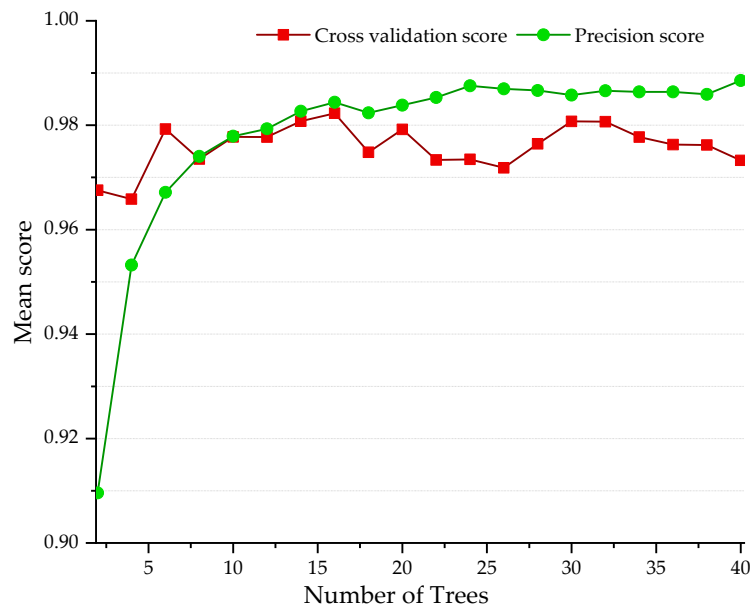


Figure 19. Mean score vs. number of trees.

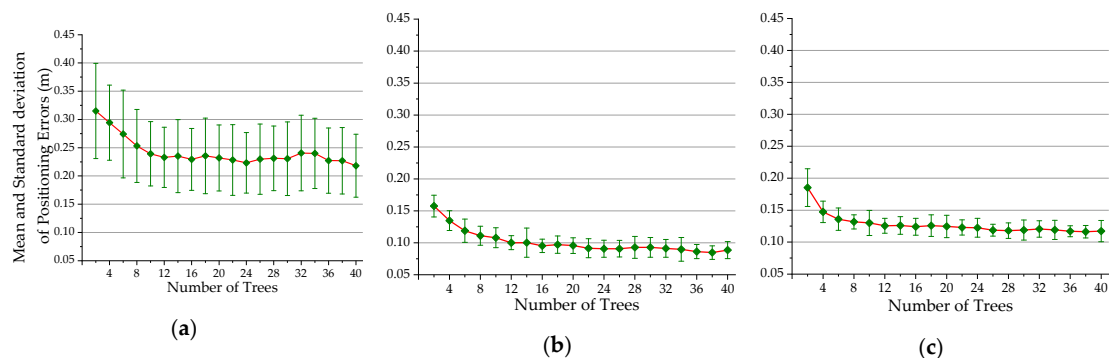


Figure 20. Mean and standard deviation of positioning errors vs. number of trees by: (a) whole floor, (b) center area, and (c) edge area.

5. Simulation and Results

5.1. Computational Time Comparison

As depicted in Figure 21, in all four algorithms, there was a significant decrease in the computational time required after applying area division, although each algorithm showed distinct levels of improvement. To compare them with each other, the specific parameters of each algorithm were first optimized. After finding the optimal parameters, the execution time for each method was calculated. The total execution time includes the time spent on the data classification process (for area division) and the location prediction process. In Figure 21, the red column shows the total CPU time before area division, while the orange and yellow columns show the total execution time for unseen data that belong to the edge area and the center area, respectively, after area division. An unknown data can only belong to either the center area or the edge area. Once this data is in the center area, the total CPU time will not include the location estimation time for the edge area data, and vice versa. Table 3 presents the exact time of each method, while the level of improvement in CPU time for each proposed algorithm is shown in Table 4.

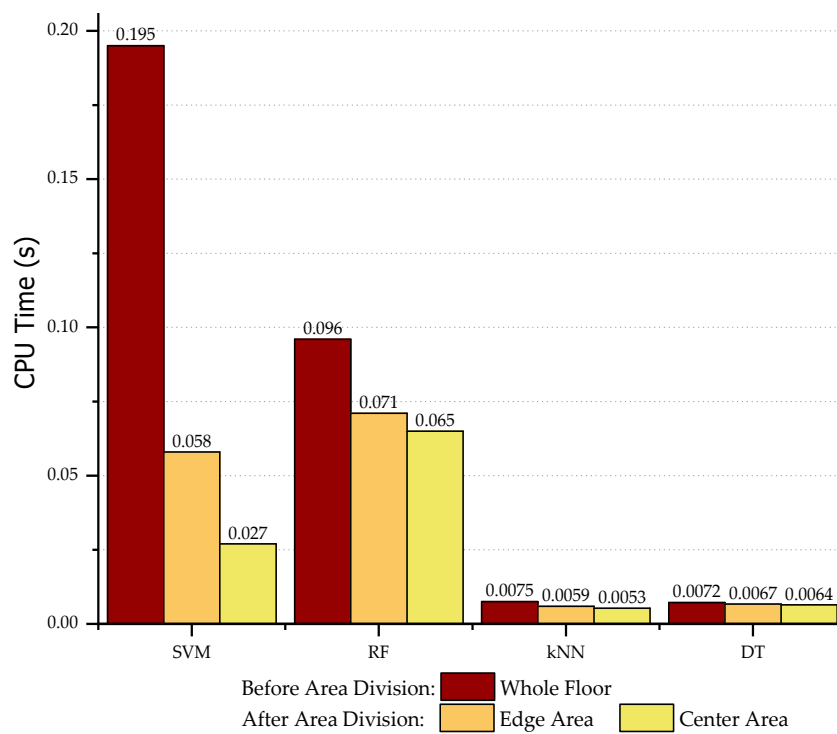


Figure 21. Computational time comparison before and after area division.

Table 3. Computational time (ms).

| Algorithms | Position of Data | Area Division Time | Location Prediction Time | TOTAL TIME |
|------------|------------------|--------------------|--------------------------|------------|
| SVM | Whole floor | No classification | 195.00 | 195.00 |
| | Center Area | 6.90 | 20.00 | 26.90 |
| | Edge Area | 6.90 | 51.00 | 57.90 |
| RF | Whole floor | No classification | 96.00 | 96.00 |
| | Center Area | 11.00 | 54.00 | 65.00 |
| | Edge Area | 11.00 | 60.00 | 71.00 |
| kNN | Whole floor | No classification | 7.50 | 7.50 |
| | Center Area | 0.59 | 4.70 | 5.29 |
| | Edge Area | 0.59 | 5.30 | 5.89 |
| DT | Whole floor | No classification | 7.19 | 7.19 |
| | Center Area | 1.10 | 5.31 | 6.41 |
| | Edge Area | 1.10 | 5.63 | 6.73 |

Table 4. CPU time level of improvement after area division (%).

| Position of Data | SVM | RF | kNN | DT |
|------------------|-------|-------|-------|-------|
| Center | 86.21 | 32.29 | 29.47 | 10.85 |
| Edge | 70.31 | 26.04 | 21.47 | 6.40 |
| Average | 78.26 | 29.17 | 25.47 | 9.63 |

It is clear that the system with SVM suffers from the heaviest computation burden but also presents the greatest improvement in execution time after area division. The improvement was 86.21% for the center area and 70.31% for the edge area (see Table 4). The kNN and DT algorithms execute much faster than RF and SVM, with total times of 7.5 ms, and 7.19 ms, respectively. Furthermore, unlike SVM, the time improvement in DT is the lowest of the four suggested methods, with 10.85% for

the center area and 6.4% for the edge area. In summary, all four methods show a significant reduction in computation cost after area division is performed using classification.

5.2. Positioning Accuracy Assessment

We performed simulations to evaluate the effectiveness of each algorithm. As shown in Figure 22, all four methods exhibited a significant decrease in positioning error after noise reduction and area division, although improvement levels varied. As seen in Table 5, DT demonstrated the most positive change, with 59.21% improvement for the center area, and 45.89% improvement for the edge area. The least improved method was kNN, which showed 32.54% improvement and 10.06% improvement for the center area and edge area, respectively. The RF and SVM algorithms also showed a very positive improvement when the average RMSE percentage was approximately the same and around 37%.

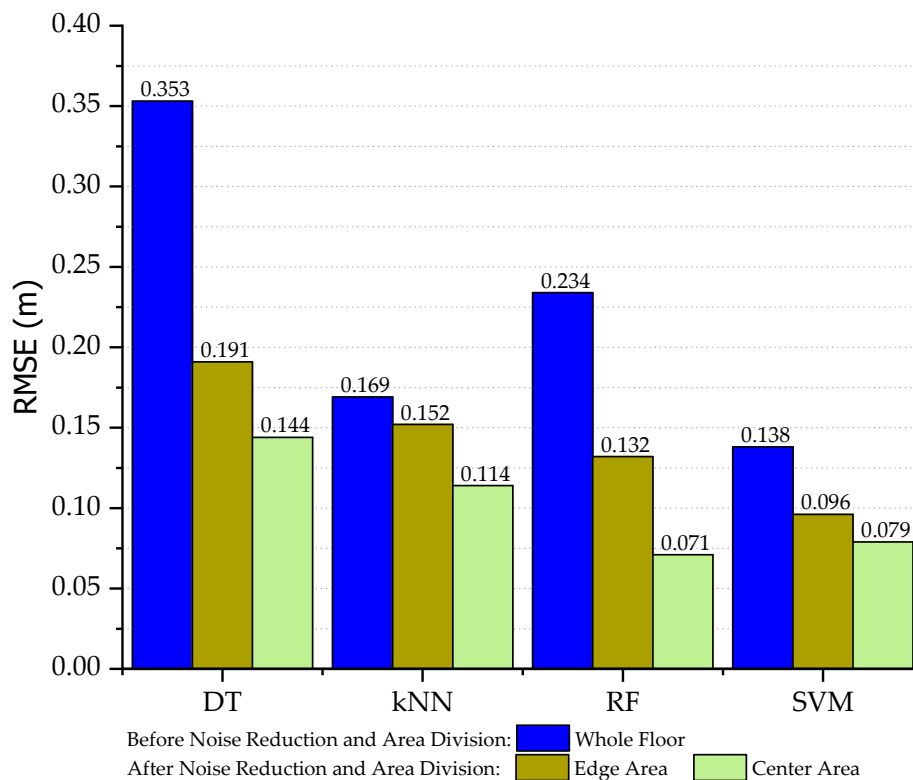


Figure 22. RMSE comparison.

Table 5. Root mean square error (RMSE) comparison (%).

| Position of Data | DT | kNN | RF | SVM |
|------------------|-------|-------|-------|-------|
| Center | 59.21 | 32.54 | 46.18 | 42.75 |
| Edge | 45.89 | 10.06 | 28.90 | 30.43 |
| Average | 52.55 | 21.30 | 37.54 | 36.59 |

To acquire a more accurate assessment of the degree of influence of each process (i.e., noise reduction and area division) on the final positioning accuracy, the RMSEs of each algorithm (i.e., DT, kNN, RF, and SVM) are surveyed under four cases (Figure 23):

- (I): Without noise reduction and without area division
- (II): Without noise reduction and with area division
- (III): With noise reduction and without area division
- (IV): With noise reduction and with area division

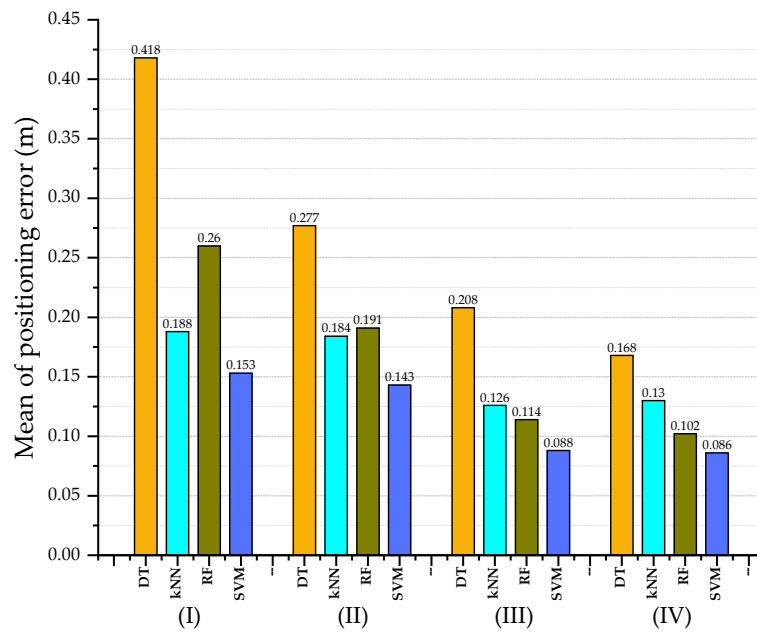


Figure 23. Detailed mean of positioning error comparison.

Case II (incorporating area division) showed a relatively small improvement in the RMSE, with the kNN and SVM algorithms showing 2.13% and 6.54% improvement, respectively. This means that area division has a negligible effect on these two algorithms in term of accuracy. In contrast, the DT and RF algorithms were strongly influenced by the area division process, showing an improvement of 33.73% and 26.54%, respectively.

Case III, with noise reduction and no area division, showed a very promising decline in positioning errors compared to Case I. SVM achieved the best improvement, followed by RF. The DT algorithm continued to show the lowest level of accuracy.

It is clear that the mean positioning errors tended to gradually improve progressing from Case I to Case IV. The highest accuracy (8.6 cm) occurs with SVM in Case IV. The worst accuracy (16.8 cm) occurs with DT. It is interesting to note, however, that the DT algorithm had the highest improvement, showing an RSME reduction of nearly 60% (from 48.1 cm in Case I to 16.8 cm in Case IV). Also, RF showed the same improvement level as DT but RF has a much better positioning accuracy of 10.2 cm. This result is only slightly worse than SVM. Of the four algorithms, kNN shows the least improvement at 30.85%.

In summary, all four methods achieve better positioning accuracy after noise reduction and area division, in the following ascending order of accuracy: DT, kNN, RF, and SVM. The area division technique had a relatively small impact on the accuracy of kNN and SVM, although it did provide significant savings in execution time, as was discussed in the previous Section.

Next, we analyzed the positioning errors, based on the hypothetical roadmap of a mobile robot. On this route, the robot started from the left upper corner, as shown in Figure 24, then went to the opposite position with LED 1 (see Figure 1), then proceeded through the center of the floor and the area near the wall. Finally, the robot went to the corner opposite the start point. Since the effects of multipath reflections and noises varied greatly from location to location in the room, this route helped us to evaluate the efficiency of the proposed solution under varying conditions.

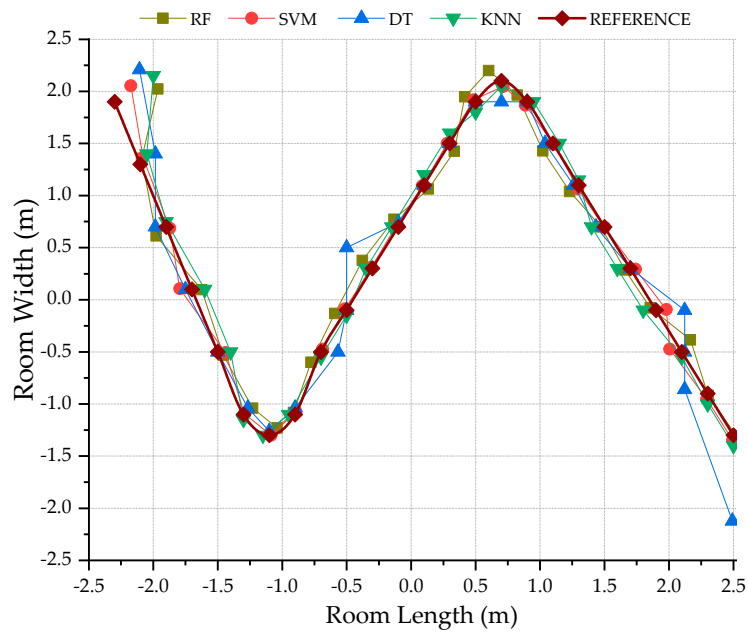


Figure 24. Estimated route before noise reduction and area division.

Before applying our proposed solution, the positioning errors in the two corners were very bad (Figure 24). A contrasting image, however, is shown in Figure 25, which shows the results after our solution is deployed. The errors significantly decreased, although the corner area still showed the least accuracy, due to multipath reflections and noises. Furthermore, positioning quality is also better expressed in the remaining areas in which the SVM and RF algorithms provided the best results. Although each method has its own advantages and disadvantages, they all showed a significant improvement in positioning accuracy after applying both area division and noise reduction.

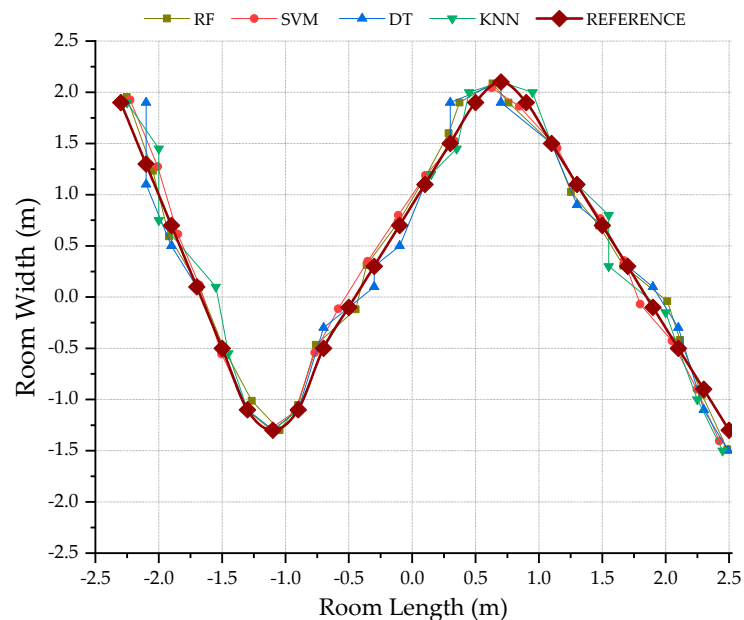


Figure 25. Estimated route after noise reduction and area division.

6. Conclusions

In summary, the simulation results showed that after applying the proposed solution, the SVM and RF methods obtained the highest positioning accuracy, at 8.6 cm and 10.2 cm, respectively. The SVM

algorithm had the best improvement in execution time (approximately 78%). Our results indicated that if practical applications need a low positioning error, SVM is the optimal option. For applications that need fast execution time but moderate accuracy, we suggest kNN, which demonstrated a positioning accuracy of 13 cm and an average computational time of 5.6 ms.

In this article, we proposed an enhanced ML-based indoor positioning solution using LED lights. Our solution not only improved positioning accuracy, but also reduced computational time. In addition to using signal pre-processing to achieve more accurate positioning, the novel adoption of dual-function ML was employed. The classification function saved execution time and provided a slight improvement in positioning accuracy, while the regression function helped determine the exact location of the object in the final step. In particular, this study compared four popular ML algorithms: SVM, DT, RF, and kNN. The results from this comparison gave us a comprehensive view of the advantages and disadvantages of each algorithm in positioning applications using VLC.

Because developing VLC-based indoor mobile robot applications is our main pursuit, the optimization of ML algorithms to improve positioning accuracy and execution time in a real environment is our highest priority. In our future work, we continue to optimize positioning accuracy using deep machine learning.

Author Contributions: H.Q.T. proposed the idea, designed the proposed algorithm, performed the simulation, and wrote the manuscript. As the corresponding author, C.H. supervised the research, provided the guidance for data analysis, and revised the article.

Acknowledgments: This work was supported by Korea Hydro & Nuclear Power company through the project “Nuclear Innovation Center for Haeoleum Alliance”.

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

References

1. Armstrong, J.; Sekercioglu, Y.A.; Neild, A. Visible light positioning: A roadmap for international standardization. *IEEE Commun. Mag.* **2013**, *51*, 68–73. [[CrossRef](#)]
2. Rajagopal, S.; Roberts, R.D.; Lim, S. IEEE 802.15.7 visible light communication: Modulation schemes and dimming support. *IEEE Commun. Mag.* **2012**, *50*, 72–82. [[CrossRef](#)]
3. *IEEE Draft Standard for Local and Metropolitan Area Networks—Part 15.7: Short-Range Optical Wireless Communications*; IEEE P802.15.7/D3; Institute of Electrical and Electronics Engineers (IEEE): Englewood, CO, USA, 1 January 2018; pp. 1–412.
4. Keskin, M.F.; Sezer, A.D.; Gezici, S. Localization via Visible Light Systems. *Proc. IEEE* **2018**, *106*, 1063–1088. [[CrossRef](#)]
5. Indoor Positioning: Perfect Light, Precise Location. Available online: <http://www.lighting.philips.com/main/systems/lighting-systems/indoor-positioning> (accessed on 3 February 2019).
6. Qiu, K.; Zhang, F.; Liu, M. Let the Light Guide Us: VLC-Based Localization. *IEEE Robot. Autom. Mag.* **2016**, *23*, 174–183. [[CrossRef](#)]
7. Murai, R.; Sakai, T.; Kawano, H.; Matsukawa, Y.; Kitano, Y.; Honda, Y.; Campbell, K.C. A novel visible light communication system for enhanced control of autonomous delivery robots in a hospital. In Proceedings of the 2012 IEEE/SICE International Symposium on System Integration (SII), Fukuoka, Japan, 16–18 December 2012; pp. 510–516. [[CrossRef](#)]
8. Zhuang, Y.; Hua, L.; Qi, L.; Yang, J.; Cao, P.; Cao, Y.; Wu, Y.; Thompson, J.; Haas, H. A Survey of Positioning Systems Using Visible LED Lights. *IEEE Commun. Surv. Tutor* **2018**, *20*, 1963–1988. [[CrossRef](#)]
9. Do, T.-H.; Yoo, M. An in-Depth Survey of Visible Light Communication Based Positioning Systems. *Sensors* **2016**, *16*, 678. [[CrossRef](#)]
10. Yang, C.; Shao, H. WiFi-based indoor positioning. *IEEE Commun. Mag.* **2015**, *53*, 150–157. [[CrossRef](#)]
11. Akram, B.A.; Akbar, A.H.; Shafiq, O. HybLoc: Hybrid Indoor Wi-Fi Localization Using Soft Clustering-Based Random Decision Forest Ensembles. *IEEE Access* **2018**, *6*, 38251–38272. [[CrossRef](#)]
12. Li, Z.; Braun, T.; Zhao, X.; Zhao, Z.; Hu, F.; Liang, H. A Narrow-Band Indoor Positioning System by Fusing Time and Received Signal Strength via Ensemble Learning. *IEEE Access* **2018**, *6*, 9936–9950. [[CrossRef](#)]

13. Faragher, R.; Harle, R. Location Fingerprinting with Bluetooth Low Energy Beacons. *IEEE J. Sel. Areas Commun.* **2015**, *33*, 2418–2428. [[CrossRef](#)]
14. Zou, H.; Wang, H.; Xie, L.; Jia, Q. An RFID indoor positioning system by using weighted path loss and extreme learning machine. In Proceedings of the 2013 IEEE 1st International Conference on Cyber-Physical Systems, Networks, and Applications (CPSNA), Taipei, Taiwan, 19–20 August 2013; pp. 66–71. [[CrossRef](#)]
15. Lin, B.; Ghassemlooy, Z.; Lin, C.; Tang, X.; Li, Y.; Zhang, S. An Indoor Visible Light Positioning System Based on Optical Camera Communications. *IEEE Photonics Technol. Lett.* **2017**, *29*, 579–582. [[CrossRef](#)]
16. Lei, Q.; Zhang, H.; Sun, H.; Tang, L. Fingerprint-Based Device-Free Localization in Changing Environments Using Enhanced Channel Selection and Logistic Regression. *IEEE Access* **2018**, *6*, 2569–2577. [[CrossRef](#)]
17. Zhang, J.; Xiao, W.; Zhang, S.; Huang, S. Device-Free Localization via an Extreme Learning Machine with Parameterized Geometrical Feature Extraction. *Sensors* **2017**, *17*, 879. [[CrossRef](#)] [[PubMed](#)]
18. Guo, X.; Shao, S.; Ansari, N.; Khreishah, A. Indoor Localization Using Visible Light via Fusion of Multiple Classifiers. *IEEE Photonics J.* **2017**, *9*, 1–16. [[CrossRef](#)]
19. Yuan, T.; Xu, Y.; Wang, Y.; Han, P.; Chen, J. A Tilt Receiver Correction Method for Visible Light Positioning Using Machine Learning Method. *IEEE Photonics J.* **2018**, *10*, 1–12. [[CrossRef](#)]
20. Xie, C.; Guan, W.; Wu, Y.; Fang, L.; Cai, Y. The LED-ID Detection and Recognition Method Based on Visible Light Positioning Using Proximity Method. *IEEE Photonics J.* **2018**, *10*, 1–16. [[CrossRef](#)]
21. Tran, H.Q.; Ha, C. Fingerprint-Based Indoor Positioning System Using Visible Light Communication—A Novel Method for Multipath Reflections. *Electronics* **2019**, *8*, 63. [[CrossRef](#)]
22. Komine, T.; Nakagawa, M. Fundamental analysis for visible-light communication system using LED lights. *IEEE Trans. Consum. Electron.* **2004**, *50*, 100–107. [[CrossRef](#)]
23. Gfeller, F.R.; Bapst, U. Wireless in-house data communication via diffuse infrared radiation. *Proc. IEEE* **1979**, *67*, 1474–1486. [[CrossRef](#)]
24. Sadowski, S.; Spachos, P. RSSI-Based Indoor Localization with the Internet of Things. *IEEE Access* **2018**, *6*, 30149–30161. [[CrossRef](#)]
25. Ghassemlooy, Z.; Popoola, W.; Rajbhandari, S. *Optical Wireless Communications, System and Channel Modeling with MATLAB*; CRC Press: Boca Raton, FL, USA, 2012; ISBN 9781439851883.
26. Mohammed, N.; Elkarim, M. Exploring the effect of diffuse reflection on indoor localization systems based on RSSI-VLC. *Opt. Express* **2015**, *23*, 20297–20313. Available online: <https://www.osapublishing.org/oe/abstract.cfm?uri=oe-23-16-20297> (accessed on 22 February 2019). [[CrossRef](#)] [[PubMed](#)]
27. Lu, X.; Zou, H.; Zhou, H.; Xie, L.; Huang, G. Robust Extreme Learning Machine With its Application to Indoor Positioning. *IEEE Trans. Cybern.* **2016**, *46*, 194–205. [[CrossRef](#)] [[PubMed](#)]
28. Gu, W.; Aminikashani, M.; Deng, P.; Kavehrad, M. Impact of Multipath Reflections on the Performance of Indoor Visible Light Positioning Systems. *J. Lightw. Technol.* **2016**, *34*, 2578–2587. [[CrossRef](#)]
29. Boucouvalas, A.C. Indoor ambient light noise and its effect on wireless optical links. *IEE Proc. J. Optoelectron.* **1996**, *143*, 334–338. [[CrossRef](#)]
30. Downey, A.B. *Think DSP—Digital Signal Processing in Python*; O’Reilly: Sebastopol, CA, USA, 2016; ISBN 978-1-491-93845-4.
31. Tagare, P. Signal averaging. In *Biomedical Digital Signal Processing*; Tompkins, W.J., Ed.; Prentice-Hall, Inc.: Upper Saddle River, NJ, USA, 1993; pp. 184–192.
32. Alin, C.; Barthélemy, C.; Luc, C.; Valentin, P.; Mihai, D. Evaluation of the noise effects on visible light communications using Manchester and Miller coding. In Proceedings of the 2014 International Conference on Development and Application Systems (DAS), Suceava, Romania, 15–17 May 2014; pp. 85–89.
33. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Vanderplas, J.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
34. Friedman, J.; Hastie, T.; Tibshirani, R. The Elements of Statistical Learning. In *Data Mining, Inference and Prediction*; Springer: New York, NY, USA, 2008; ISBN 978-0387848570.
35. Ben-Hur, A.; Weston, J. *A User’s Guide to Support Vector Machines. Data Mining Techniques for the Life Sciences*; Springer: New York, NY, USA, 2010; pp. 223–239. [[CrossRef](#)]
36. James, G.; Witten, D.; Hastie, T.; Tibshirani, R. *An Introduction to Statistical Learning with Application in R*; Springer: New York, NY, USA, 2014; ISBN 978-1461471370.

37. Bramer, M. Avoiding Overfitting of Decision Trees. In *Principles of Data Mining. Undergraduate Topics in Computer Science*; Springer: London, UK, 2013. [[CrossRef](#)]
38. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5. [[CrossRef](#)]



© 2019 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 (<http://creativecommons.org/licenses/by/4.0/>).