Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

ARTICLE TYPE

Exploiting Temporal Dependency of RSS Data with Deep Learning for IoT-oriented Wireless Indoor Localization

Qianwen Ye[1,2] | Xiaochen Fan*[3] | Gengfa Fang[2] | Hongxia Bie[1]

[1]School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing, China
[2]School of Electrical and Data Engineering, University of Technology Sydney, NSW, Australia
[3]Centre for Assessment and Demonstration Research, AMS, Beijing, China

Correspondence
*Xiaochen Fan, No.1 Xianghongqi, Haidian District, Beijing, 100091, China.
Email: fanxiaochen33@gmail.com

With the ubiquitous demand for indoor location-based services and the pervasive deployment of Wi-Fi hotspots, wireless indoor localization has been widely studied by utilizing various Wi-Fi signal measurements. Most existing schemes leverage the Received Signal Strength (RSS) of Wi-Fi to conduct cost-efficient indoor localization. However, the RSS data are not only prone to multi-path effects, but also sensitive to time-varying environmental dynamics, making it quite daunting to achieve robust indoor localization. In contrast to existing solutions that focus on spatial features of RSS, in this article, we exploit the temporal dependency of RSS time-series data by integrating the Kalman filter with deep neural networks. In particular, to tame time-varying noises and preserve valuable temporal features in RSS measurements, we propose a time-varying RSS filtering algorithm based on the Kalman filter and a refined post-processing module. Moreover, a deep learning model based on deep neural network (DNN) is further utilized for effective feature extraction on one-dimension RSS fingerprints. The experiment results show that the proposed Kalman-DNN model improves at least 25% localization accuracy in comparison with conventional DNN model. Furthermore, with the localization time as 0.02 millisecond (ms), the Kalman-DNN model outperforms the Kalman-CNN model in localization accuracy by at least 10%.

KEYWORDS:
Intelligent Signal Processing, Deep Learning, Internet of Things, Indoor Localization

1 | INTRODUCTION

With the proliferation of portable Internet of Things (IoT) devices and the penetration of wireless networks in public indoor space, indoor localization has become imperative to support a variety of indoor location-based services. To facilitate wireless indoor localization, researchers have explored many short-range communication signals, including ZigBee, Bluetooth, Wi-Fi, cellular networks, as well as their combinations [1]. Due to its wide availability and ubiquitous accessibility, the Wi-Fi access point (AP) has become one of the most attractive infrastructures for indoor localization. In particular, the received signal strength (RSS) of Wi-Fi has received intensive research interests from research community in achieving none-line-of-sight indoor localization. However, the RSS-based indoor localization is inherently vulnerable to multi-path effects and environmental dynamics, which bring signal reflections and even signal fading [2]. The above vulnerabilities may significantly compromise the efficiency and accuracy of Wi-Fi RSS-based indoor localization.

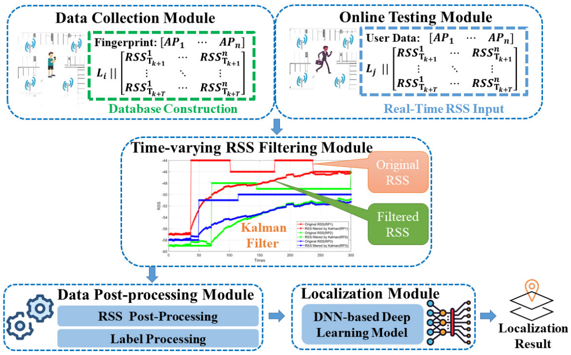


FIGURE 1 The system architecture of Kalman-DNN.

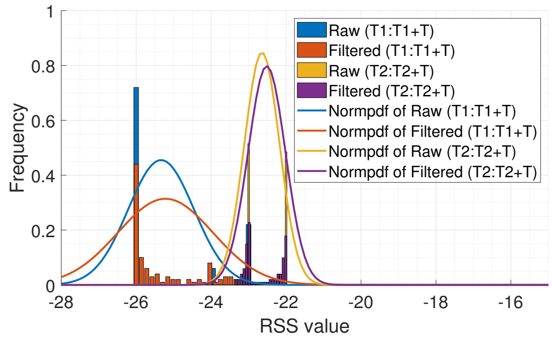


FIGURE 2 The RSS time-series and the corresponding Normally distributed probability density function (Normpdf) before/after Kalman filtering.

Existing efforts mainly focused on taming external influence on RSS to enhance the performance of indoor localization. For instance, He *et al.* proposed a Gaussian regression model to compensate for frequency-dependent shadowing effects and multi-path in RSS[3]. To reduce error mitigation, Katwe *et al.* presented an effective hybridization measurement of time of arrival and RSS[4]. While the existing schemes can improve localization accuracy in some typical indoor scenarios, the fundamental limit of RSS's continuous dependency on environmental dynamics is still not fully addressed yet[5]. To overcome the above limitations, RSS fingerprinting has been extensively adopted to enable efficient data collection for radio map construction.

In this letter, we explore the temporal dependency of RSS data through combining the Kalman filter with deep learning. We aim to achieve effective RSS signal processing and devise an RSS filtering algorithm with the Kalman filter. After that, a post-processing module is further leveraged to compress the RSS time-series data and reduce the computation complexity, as it transforms the original RSS inputs into dynamic-resistant RSS time-series fingerprints. We further employ a deep neural network (DNN) to extract useful representations from RSS time-series for localization model training. We conduct extensive experimental studies on a real-world indoor localization testbed, and the results show that it is worth combining Kalman filtering algorithm with deep learning to process RSS measurements for IoT-oriented indoor localization.

The rest of this letter is organized as follows. Section II introduces the system architecture of the proposed indoor localization system. Section III elaborates the algorithm design and the DNN model of Kalman-DNN that exploits the temporal dependency of RSS data. Section IV presents evaluation results. At last, Section V concludes this letter and provides an outlook of our future work.

2 | OVERVIEW OF KALMAN-DNN SYSTEM

As shown in Fig. 1, the system architecture of Kalman-DNN consists of 5 modules, *i.e.*, the data collection module, the RSS filtering module, the data post-processing module, the online testing module and the localization module. Different from existing studies, we exploit the temporal dependency of fingerprints by utilizing RSS time-series collection for feature extraction and model training. We denote each reference point (RP) as $L_i(i = 1, 2 \dots I)$ and use $\{RSS^n_{T_k}, \dots, RSS^n_{T_k+t}\}$ to represent raw RSS time-series measured from n_{th} AP across time slot T_k to $T_k + t$, where $k \in \mathbb{N}_+$.

During the offline training procedure, the raw RSS time-series are pre-processed by the time-varying RSS filtering algorithm to exclude noises caused by multi-path effects and environmental dynamics. Then, the filtered RSS time-series are further processed and compressed by the post-processing module to decrease the computation complexity. After that, the DNN model learns temporal features from RSS time-series fingerprints with corresponding labeled locations. For the online testing phase, we utilize raw RSS time-series collected by mobile users for localization. The main steps of RSS filtering and post-processing are the same as the offline phase. Note that the trained DNN model with optimized parameters is employed to compute the similarity between the RSS time-series measurements to the radio maps in the offline database. Finally, the Kalman-DNN outputs localization results preserve the best-match with the RSS inputs.

Algorithm 1 The Time-varying RSS Filtering Algorithm

Require: The measured RSS value of the current state at time t: $\boldsymbol{y}(t)$; The estimated RSS value of the previous state at time t-1: $\widetilde{\boldsymbol{x}}(t-1)$; The smoothed estimated error covariance of the previous state at time t-1: $\widetilde{\boldsymbol{P}}(t-1)$; The system process variance matrix: $\boldsymbol{Q}$; The system noise covariance matrix: $\boldsymbol{R}$;

Ensure: The smoothed estimated RSS value of the current state at time t: $\boldsymbol{x}(t)$; The smoothed estimated error covariance at time t: $\widetilde{\boldsymbol{P}}(t)$;

1: Predict the prior estimate of RSS value at time t: $\widetilde{\boldsymbol{x}}(t|t-1)$ by Equation 3;
2: Predict the prior estimate of error covariance at the time t: $\widetilde{\boldsymbol{P}}(t|t-1)$ by Equation 4;
3: Predict the gain at time t: $K(t)$ by Equation 5;
4: Predict the smoothed estimated error covariance at time t: $\widetilde{\boldsymbol{P}}(t)$ by Equation 6;
5: Predict the smoothed estimated RSS value at time t: $\widetilde{\boldsymbol{x}}(t)$ by Equation 7;
6: **return** $\widetilde{\boldsymbol{x}}(t)$, $\widetilde{\boldsymbol{P}}(t)$;

3 | RSS FILTERING, DATA PROCESSING AND MODEL TRAINING

3.1 | Time-varying RSS Filtering Algorithm

As shown in Fig. 2, for two different time moments T_1 and T_2, we sample T consecutive RSS time-series data, respectively. It can be observed from RSS time-series data that there are different types of sudden fluctuations in RSS values at T_1 and T_2. Caused by environmental dynamics, such random and abrupt changes in RSS measurements can significantly compromise the accuracy of indoor localization. In this article, we aim to tame such fluctuations and propose a time-varying RSS filtering algorithm based on the Kalman filter. To begin with, the Kalman filter-based algorithm will take the measured RSS time-series of the current state (*i.e.*, a period of T) as the input, which is denoted by $\boldsymbol{y}(t) = (RSS_t^1, RSS_t^2, \ldots RSS_t^n)$ and $(t = T_k + 1, T_k + 2 \ldots T_k + T)$. Then, for RSS time-series data in the previous state, we denote its error covariance matrix as $\widetilde{\boldsymbol{P}}(t-1)$ and utilize it to predict the smoothed values of estimated RSS time-series of the current state $\widetilde{\boldsymbol{x}}(t) = (\widetilde{RSS}_t^1, \widetilde{RSS}_t^2, \ldots \widetilde{RSS}_t^n)$.

To this end, the state space model for the proposed Kalman filter can be written as

$$\boldsymbol{x}(t) = \boldsymbol{x}(t-1) + w(t-1), \tag{1}$$

$$\boldsymbol{y}(t) = \boldsymbol{x}(t) + v(t), \tag{2}$$

where $\boldsymbol{x}(t)$ and $\boldsymbol{y}(t)$ are state and measurement variances, respectively. $w(t)$ and $v(t)$ are the system noise and observation noise with covariance matrices $\boldsymbol{Q}_n$ and $\boldsymbol{R}_n$, respectively.

We further combine the system noise and observation noise to calculate the estimated RSS time-series of the current state with the following equations:

$$\widetilde{\boldsymbol{x}}(t|t-1) = \widetilde{\boldsymbol{x}}(t-1), \tag{3}$$

$$\widetilde{\boldsymbol{P}}(t|t-1) = \widetilde{\boldsymbol{P}}(t-1) + \boldsymbol{Q}, \tag{4}$$

$$K(t) = \widetilde{\boldsymbol{P}}(t|t-1)(\widetilde{\boldsymbol{P}}(t|t-1) + \boldsymbol{R})^{-1}, \tag{5}$$

$$\widetilde{\boldsymbol{P}}(t) = (1 - \boldsymbol{K}(t))\widetilde{\boldsymbol{P}}(t|t-1), \tag{6}$$

$$\widetilde{\boldsymbol{x}}(t) = \widetilde{\boldsymbol{x}}(t|t-1) + K(t)(\boldsymbol{y}(t) - \widetilde{\boldsymbol{x}}(t|t-1)), \tag{7}$$

where $\widetilde{\boldsymbol{x}}(t|t-1)$ and $\widetilde{\boldsymbol{P}}(t|t-1)$ represent the posteriori state estimate and the error covariance matrix at time t, given measurements until time $t-1$. $\widetilde{\boldsymbol{x}}(t-1)$ and $\widetilde{\boldsymbol{P}}(t-1)$ represent the posteriori state estimate and the error covariance matrix at time $t-1$, given measurements until time $t-1$. $\boldsymbol{K}(t)$ is the Kalman gain, $\boldsymbol{Q}$ and $\boldsymbol{R}$ are the covariances of process and measurement noise, respectively.

Based on the above mathematical equations of a basic linear Kalman filter, we further devise the time-varying RSS filtering algorithm and its pseudo-code in Algorithm 1. We preset the initial error covariance matrix $\boldsymbol{P}(0)$ as [1], the noise covariance matrix $\boldsymbol{Q}$ as [0.001] and the observed noise covariance matrix $\boldsymbol{R}$ as [0.1].

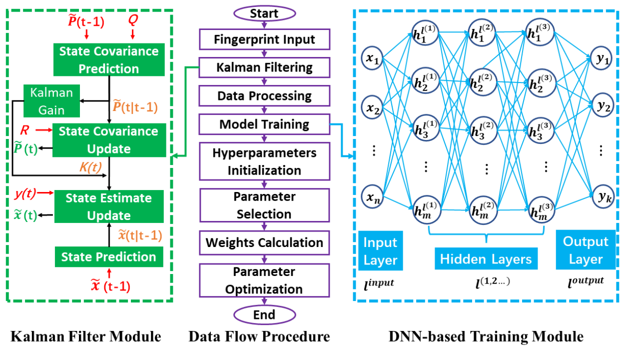


FIGURE 3 The flowchart of the offline training phase and the architecture and parameters of the DNN model.

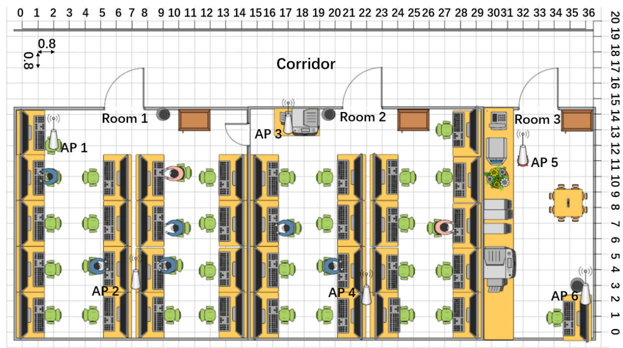


FIGURE 4 The floor plan of the experiment field.

3.2 | Data Post-processing

3.2.1 | RSS Data Post-processing

After the time-varying RSS filtering algorithm, we further introduce the post-processing module, which is designed to compress the RSS and reduce the computation complexity. First, we derive the mean value of each set of T RSS time-series samples. Then, we normalize the calculated RSS values by

$$r_i = \begin{cases} 0 & RSS_i \text{ is none,} \\ 0.1 * (RSS_i - \min) & \text{otherwise,} \end{cases} \tag{8}$$

where r_i is the normalized RSS value from AP i, RSS_i is the raw RSS value from AP i, and min is the smallest RSS value in all the averaged RSS measurements.

3.2.2 | Label Processing

To determine the label of RSS fingerprints at each reference point, we divide the localization area into a number of zones. Each zone is a grid area covering $1.6 \times 1.6\ m^2$. To generate the label for each grid, we adopt One-Hot Encoding [6] to map each grid into a One-Hot vector. Consequently, each individual grid represents a categorical variable and the indoor localization task essentially becomes a classification problem across all grids with ground-truth fingerprints.

3.3 | DNN Model Training

As shown in Fig. 3, the Kalman-DNN model consists of a multi-layered DNN model with multiple hidden layers. The proposed DNNs consist of three types of layers, including the input layer, the hidden layers, and the output layer. Based on the output of the previous layer, a non-linear function of hidden layers is as follows.

$$\boldsymbol{h}^{l^{(i)}} = f(W^{l^{(i)}} \boldsymbol{h}^{l^{(i-1)}} b^{l^{(i)}}), \tag{9}$$

where $W^{l^{(i)}}$ is the matrix of weights, indicating all the synaptic connections between each neuron of layer $l^{(i-1)}$. Each h neuron of layer $l^{(i)}$, $b^{l^{(i)}}$ is the bias vector of layer $l^{(i)}$, $\boldsymbol{h}^{l^{(i-1)}}$ is the output of the previous layer $l^{(i-1)}$, and $f(\cdot)$ is the activation function that calculates the non-linear relationship between layers.

Fig. 3 presents the flowchart of the offline training phase for indoor localization and the parameters of the DNN model. For parameter tuning, we conduct a grid search to find the best parameters to improve localization accuracy. We also train DNN models with different parameter settings for comparison purposes. We choose the rectified linear (*i.e.*, RELU) function as the activation function for the input and hidden layers. The output unit's activation function is the softmax and the loss function is the categorical cross-entropy. In our model training, we employ Adam as the optimizer of the proposed Kalman-DNN model.

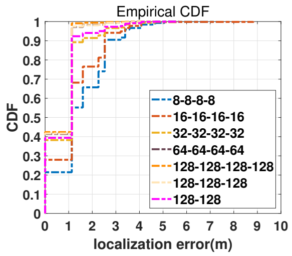


FIGURE 5 The localization errors by Kalman-DNN with different parameter settings.

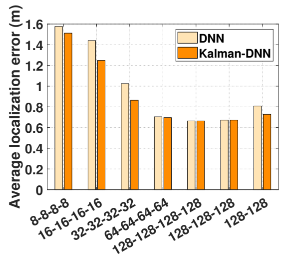
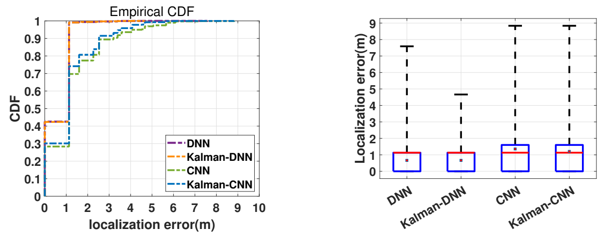


FIGURE 6 Performance comparison of DNN and Kalman-DNN.



(a) CDF **(b)** Box-plot

FIGURE 7 The overall performance comparisons among Kalman-DNN and the baseline methods.

4 | EXPERIMENTAL STUDY

We implement a real-world indoor localization testbed in the IoT lab of Beijing University of Posts and Telecommunications, as illustrated in Fig. 4. In this experimental environment, we deploy 6 Wi-Fi APs to cover three lab rooms along with a corridor area (totally 460 m^2). To achieve cost-efficient localization, we place 2 TP-Link wireless APs in each room and set a number of reference points that are evenly distributed across each room. The distance between two adjacent RPs is 0.8m and we measure RSS fingerprints at each RP for 300 times. The final dataset contains over 33,600 fingerprinting samples, with 60% fingerprints as the training set and the rest 40% as testing set. The proposed Kalman Filter algorithm and DNN model are implemented in the Tensor Flow framework, using a Dell laptop with Intel Core i7-7600 CPU.

4.1 | The Effect of DNN Parameters on Localization Errors

Fig. 5 shows the Cumulative Distribution Function (CDF) of localization errors by Kalman-DNN models with different parameter combinations. For instance, the model parameter 128-128-128-128 indicates a Kalman-DNN model that has 4 fully connected layers with 128 filters in each layer. Note that the final output layer of the Kalman-DNN is a fully connected neural network layer. From Fig. 5, we can observe that the localization accuracy is positively correlated to the number of filters in each layer. For instance, when the model parameter is 128-128-128-128, the localization errors for over 99% of testing data are under 2 m. Moreover, with the same number of filters, when the number of layers is reduced from 5 to 3, the localization accuracy significantly drops (*e.g.*, the localization errors are larger than 2 m for over 8% of the testing results).

4.2 | Experiment Result of the RSS Filtering Algorithm

Recall from Fig. 2 that T RSS time-series starting from $T1$ and $T2$ are different with noises, especially when confronting abrupt fluctuations caused by environmental dynamics. We leverage a Kalman-based filtering algorithm to tame the above noises and fluctuations and the relevant experimental results are shown in Fig. 6. We compare the average localization errors under the conventional DNN model and the proposed Kalman-DNN model. In particular, when model parameters are set as 8-8-8-8, 16-16-16-16, 32-32-32-32, 64-64-64-64 and 128-128, our Kalman-DNN consistently outperforms the basic DNN model by reducing 0.07 m, 0.2 m, 0.16 m, 0.01 m and 0.08 m in average localization errors, respectively. Since our testbed with only 6 APs accounts for small-scale localization scenarios, the integration of Kalman-DNN has already made a remarkable difference in improving localization accuracy. We also find more significant improvement of Kalman-DNN to conventional DNN in experiments with large-scale indoor localization datasets, such as UJIIndoorLoc dataset and Tampere dataset[7]. However, due to the page limit, we omit the above experimental results in this article.

4.3 | Performance Comparison of Different Localization Algorithms

To further evaluate the performance of Kalman-DNN that integrates temporal RSS features with deep neural networks, we compare the localization performance of DNN, CNN, Kalman-DNN and Kalman-CNN in Fig. 7. In this experiment, we set the mean localization (*i.e.*, the mean computation time for an RSS fingerprint input in online testing) as 0.02 milliseconds. The CNN and Kalman-CNN models both have three layers, with 16 filters for convolutional operations. Similarly, the DNN and Kalman-DNN models both have three layers, with 128 filters in each layer. First, as revealed from Fig. 7a, the Kalman-DNN achieves the best performance with over 99% of localization errors smaller than 1.13 m. By exploiting temporal dependency of RSS measurements, the Kalman-DNN model improves at least 25% and 10% accuracy in comparison with the conventional DNN model and the Kalman-CNN model, respectively. Second, we further compare their performance in a box-plot in Fig. 7b. Overall, Kalman-based deep learning models achieve better localization results with smaller errors than conventional deep learning models. For instance, the Kalman-CNN improves the medium localization error by 0.2 m in comparison with conventional CNN model. The above experimental results show that the Kalman Filters can effectively tame the noises in RSS time-series and improve localization accuracy. By exploiting temporal features of RSS measurements, the proposed Kalman-DNN model achieves the best performance among all baseline methods.

5 | CONCLUSION

In this article, we have proposed a novel indoor localization method utilizing the temporal dependency of RSS data with DNN for IoT-oriented wireless indoor localization. The Kalman Filter-based RSS filtering algorithm is leveraged to tame the random noises in RSS time-series data. To further reduce the computation complexity, a post-processing module has been proposed together with a label processing module. To efficiently extract robust features, a DNN-based deep learning model is further applied with online training. Extensive field experiments have been conducted using a real-world testbed, and the experimental results validate the effectiveness of the proposed Kalman-DNN model. Overall, the Kalman-DNN model can improve up to 25% localization accuracy in comparison with the conventional DNN model. In addition, the Kalman-DNN model outperforms the Kalman-CNN model by 10% in localization accuracy with the mean localization time as 0.02 ms. In future work, we will further exploit spatial and temporal features in RSS fingerprints for ubiquitous indoor localization solutions. The deep learning methods are promising in feature extraction of indoor localization. In particular, security issues are rising as new challenges that would compromise the accuracy of indoor localization, such as AP attacks. In this regard, semi-supervised deep learning methods (*e.g.*, denoise autoencoder) can be utilized to exploit spatial and temporal features of RSS data, thus taming the influence of AP attacks and enhancing the security of indoor localization.

CONFLICT OF INTEREST

The authors declare no potential conflict of interests.

References

1. Xiao J, Zhou Z, Yi Y, Ni LM. A survey on wireless indoor localization from the device perspective. *ACM Computing Surveys (CSUR)* 2016; 49(2): 25.
2. Zhu X, Qu W, Qiu T, Zhao L, Atiquzzaman M, Wu DO. Indoor Intelligent Fingerprint-Based Localization: Principles, Approaches and Challenges. *IEEE Communications Surveys & Tutorials* 2020; 22(4): 2634–2657.
3. He Z, Li Y, Pei L, Chen R, El-Sheimy N. Calibrating multi-channel RSS observations for localization using Gaussian process. *IEEE Wireless Communications Letters* 2019; 8(4): 1116–1119.
4. Katwe M, Ghare P, Sharma PK, Kothari A. NLOS Error Mitigation in Hybrid RSS-TOA-Based Localization Through Semi-Definite Relaxation. *IEEE Communications Letters* 2020; 24(12): 2761–2765.
5. Tian X, Wang M, Li W, et al. Improve Accuracy of Fingerprinting Localization with Temporal Correlation of the RSS. *IEEE Transactions on Mobile Computing* 2018; 17(1): 113-126.
6. Buckman J, Roy A, Raffel C, Goodfellow I. Thermometer Encoding: One Hot Way To Resist Adversarial Examples. In: International Conference on Learning Representations. ; 2018.
7. Song X, Fan X, Xiang C, et al. A novel convolutional neural network based indoor localization framework with WiFi fingerprinting. *IEEE Access* 2019; 7: 110698–110709.