Future Generation Computer Systems 151 (2024) 196–213

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

An explainable hybrid deep learning architecture for WiFi-based indoor localization in Internet of Things environment

Zeynep Turgut, Arzu Gorgulu Kakisim *

Istanbul Medeniyet University, Computer Engineering, Uskudar, İstanbul, Turkey

ARTICLE INFO

Keywords:
Deep learning
Explainable AI
IoT
Particle filter

ABSTRACT

The indoor positioning service is one of the essential services needed in the Internet of Things ecosystem. Recently, many researchers have focused on the fingerprinting method, which is a method based on signal mapping with the Received Signal Strength Indicator (RSSI) values obtained from the WiFi access points. However, the fingerprinting method is particularly challenging due to some difficulties, such as RSSI variance over time, device diversity, and similarities of fingerprints in indoor networks. For this reason, machine learning and deep learning methods are used for many purposes, such as estimating the location of the building, floor, or the rooms. Detecting the location of a room or more than one reference point in a room becomes a more difficult problem because neighboring reference points' fingerprints are very similar to each other. This study proposes a WiFi-based XAI-empowered deep learning architecture to predict the reference points in a room or corridor. We present a hybrid deep learning-based method that uses Long-Short-Term Memory to capture long-term dependencies between the signal features, and Convolutional Neural Network to extract local spatial signal patterns. Our deep learning aims to enrich fingerprinting data of each sample to capture more meaningful feature maps coming from different angles. Moreover, the method applies effective filtering and dimension scaling on the data to regulate the RSS values and capture more discriminative patterns using particle filter and sparse autoencoder. To provide local and global explanations for indoor localization estimations, the proposed architecture comprises two Explainable Artificial Intelligence techniques as Interpretable Model-Agnostic Explanations, and SHapley Additive exPlanations. The experimental results demonstrate that the proposed architecture achieves higher accuracy values for all datasets than the baseline deep learning methods.

1. Introduction

The Internet of Things (IoT) is a popular communication concept that entered our lives with the promise that all objects can communicate with each other. It has rapidly realized its commitments since the first mention of Kevin Ashton in 1999 [1]. The Internet of Things ecosystem, which we encounter in many places, from logistics facilities to smart home designs, from museum visits to hospital administrations, has rapidly evolved from a network technology that makes life easier to an indispensable infrastructure. The Internet of Things is the key infrastructure of our assistants, who can inform our mobile phone of the location of our lost car key, remind the expiration dates of our products left in the refrigerator, transmit the data received from our pacemaker, and mobile phone to our doctor, and limit their services only to our dreams and needs. What an IoT (Internet of Things) ecosystem can offer is directly related to the services it contains. For this reason, the usage of many different services are needed in IoT environments with their dynamic structure that can change at any time in the ecosystem. The environment offered by services such as content management service, mobility management service, data management service, reliability service, energy management service, and location determination service brings "smartness" to every environment. Smart buildings, smart hospitals, smart schools, smart home design, and every area, including indoor areas, especially, need an indoor positioning system. The indoor positioning system will be able to track the location of patients and people in need of help in smart homes, show the shortest route to the artworks that visitors want to see from where they are in smart museums, determine the location of the product needed in a smart factory and may bring it by smart robots in a smart hospital, will be able to initiate the necessary process for the destruction of expired products. Creating indoor positioning systems is particularly challenging due to the inability to use GPS signals, which provide precise solutions in outdoor areas, and the characteristics of indoor areas, which include different approaches [2]. Indoor spaces have high

* Corresponding author.
E-mail addresses: zeynep.turgut@medeniyet.edu.tr (Z. Turgut), arzu.kakisim@medeniyet.edu.tr (A.G. Kakisim).

https://doi.org/10.1016/j.future.2023.10.003
Received 31 December 2022; Received in revised form 3 July 2023; Accepted 6 October 2023
Available online 14 October 2023

dynamics: the number of objects and people in indoor locations can change, and the signal values collected for location detection can be exposed to different effects. They can take a very different value from their original values in every time period. All the effects that may encounter during signal communication in indoor areas are challenges for indoor localization approaches.

Positioning approaches in any indoor location are generally divided into deterministic and probabilistic approaches [3–5]. Deterministic methods have different approaches such as trilateration, triangulation, proximity, and dead reckoning [6]. At least three devices must be located in an enclosed space to successfully use techniques such as trilateration, which is based on the principle of using signal timestamps from three devices, and triangulation, which is based on the principle of using the arrival angles of signal values from three devices. It may require extra hardware costs, and at the same time, the performance of these techniques is directly affected by the effects of the signal. Dead reckoning is based on the approach of estimating a user's location from the last known point, and in case of possible deviations, the error in the predicted location grows with each step. Proximity requires the presence of reference devices because it works on the principle of finding proximity to an existing signaling device. Fingerprinting method, which is one of the probabilistic methods, is a method that includes two stages, online and offline, and is based on the extraction of the signal map of the area where the location is desired to be determined [6]. In the offline phase, a signal map is created by collecting the signal values in the area to be positioned. The areas where the signal values will be collected are usually divided into areas called grids, and signal samples are taken from the middle, side, or both the middle and sides of the grids. Technologies that can be found indoors, such as WiFi, and Bluetooth Low Energy, can be used to create the signal map. In particular, WiFi technology is frequently used in fingerprinting-based studies due to the presence of WiFi access points in almost every building. Regardless of the technology used, effects such as distortions, scattering, multi-path effects, and internal noises of signal-collecting devices may cause the original signal value not to be obtained [7]. For this reason, the creation of the signal map free of effects in the offline stage is the first difficulty of fingerprint-based studies. The online stage is based on positioning by comparing the signal values obtained from the mobile device of the mobile user with the signal map created in the offline stage.

Traditionally, the main idea of RSSI fingerprint-based approaches is first to create a database of RSSI fingerprints, to implement a learning phase using generally machine learning approaches, then to assign a location estimate by matching the fingerprints reported by the user. At this stage, while some studies transform the problem into a regression problem for estimating three-dimensional location coordinates, including longitude, latitude, and altitude, some studies convert the problem to a classification problem defining each reference point as a class for the learning model [8]. Some of these methods focus on estimating building location, some on both building and floor location, and some on room location in more detail. Considering the proximity between the rooms and corridors on the floor, the problem of estimating the room location is very difficult since neighboring reference points have very similar RSSI signals. Although regression-based methods provide a more generalized approach (only) that does not consider the number of floors or buildings since they only estimate the three-dimensional location coordinates during the learning phase, it has been observed in the literature that more effective results are obtained when the indoor location estimation problem is considered as a classification problem in general [9–12]. In particular, if a smart indoor localization system is to be used for a single smart building with a known floor number (for example, a shopping mall consisting of a single building or a company), or only a single flat (smart home) in a building, the problem can be designed as a classification problem.

The fingerprinting-based localization has some challenges [13]. During the fingerprinting phase, RSSI values fluctuate over time due to non-line-of-sight propagation or environmental dynamics, and the RSSI variance may differ over time. Since different devices may have different technological designs, they may not have the same propagation power. Therefore, RSS readings may differ from each other. In the indoor environment, walls, furniture, and living things can greatly reduce the WiFi signals. The strength of the signals that are expected to be discriminative at some points may not be obtained at the desired level. Moreover, similar RSS values are frequently observed at different reference points in the same room or the same corridor. Therefore, fingerprints of neighboring reference points are difficult to distinguish. To deal with these difficulties, recently, some researchers have focused on estimating indoor positions using deep learning-based methods [9,14] that apply deep feature engineering to capture the most relevant and meaningful patterns by mapping the data to the latent spaces. These models aim to learn low-level latent representations that comprise local discriminative signal patterns as well as long-term correlations between time steps of signal samples. To increase the performance of deep learning models, many methods utilize a feature extraction process as data normalization or filtering. The studies demonstrate that the filtering and normalization steps effectively obtain higher accuracy [15,16]. Moreover, to reveal more meaningful patterns, many existing models [10,17] use an autoencoder at a layer earlier than deep learning architectures such as Convolutional Neural Network (CNN) [18,19] and Long Short Term Memory Networks (LSTM) [20]. While the usage of these deep learning methods is certainly not new, developing new architectures that can achieve higher accuracy performance in estimating indoor locations compared to basic deep learning architectures is still an open problem.

In this study, we present a new XAI-empowered hybrid deep architecture for a WiFi-based indoor localization system on WiFi fingerprinting. Our model defines each grid that is determined by fingerprinting as a separate class and transforms the indoor localization problem as a localization classification problem. Our model is capable of predicting the location of reference points independently of floors. The method runs the proposed system in parallel to determine the floor number. It returns the specified reference point and floor number to the user. The model uses two different main layers simultaneously, which are Long-Short-Term Memory (LSTM) to capture long-term dependencies between the signal features, and a convolutional layer to extract local spatial signal patterns. Different hidden representations are produced for the same dataset with different deep learning layers, CNN and LSTM, which have different expression capabilities. With the deep feature fusion approach, the model combines and enriches the different feature maps obtained. To capture the more meaningful and distinctive low-dimensional representations from the enriched feature maps, the proposed method applies a second convolutional layer and pooling layer. The proposed method implements a main feature extraction phase using a particle filter and sparse autoencoder (SAE) for eliminating potential noise in data, revealing discriminative representations of the samples, and dimension scaling. To interpret and explain indoor position predictions generated by our deep learning model, two different XAI approaches: Interpretable Model-Agnostic Explanations (LIME) and SHapley Additive exPlanations (SHAP), are integrated into the proposed architecture. By implementing LIME, our system aims to provide detailed information about the location where new incoming data is allocated, the distribution of the prediction probabilities, and the feature distribution belonging to WiFi signals that play a role in this estimation. The SHAP model is leveraged to provide global interpretations about the feature distributions, which our deep learning model mainly uses in determining reference points. The main contributions of this work are presented as follows:

- The proposed model uses a new hybrid deep feature fusion model that implements CNN and LSTM on the same data simultaneously to obtain different representations of the data and uses a final convolutional and pooling layer to capture distinctive

low-dimensional representations of WiFi data. The model reveals long-term dependencies between signal properties with LSTM, while capturing local distinctive patterns with CNN. It provides data enrichment with feature maps coming from both different angles, thus enabling us to reveal more distinctive patterns with simultaneous deep feature engineering. Therefore, for reference points in the same room or same corridor with similar RSSI values, the model gains the ability to capture low variations in WiFi signals.

- Our deep architecture implements a sparse autoencoder before the deep feature fusion learning model. By applying a bottleneck, SAE aims to focus on extracting the compressed information representation of the original input and eliminate potential noise and redundant and irrelevant features within the data. In the study, apart from the advantage of SAE to capture the more meaningful and distinctive part of the data, SAE was also used to apply size expansion and reduction in the feature space. Deep learning architectures may not be expected to work successfully in indoor environments with a limited number of access points (APs), for example, in an indoor with only 5 APs, which means a limited-length vector representation, for example considering the convolutional operations. On the other hand, location detection in indoors with a large number of APs entails higher computational costs. In the proposed model, by using the feature representations taken from the bottleneck layer of SAE, For interiors with a relatively small number of APs, the feature space is expanded, while for indoor areas with a large number of APs, the feature space is reduced in dimension.
- Our method utilizes an effective filtering phase on WiFi signals using particle filter to regulate the RSSI values that fluctuate over time, which cannot be obtained with the same propagation power due to different technological designs, and which are obtained at a very low frequency than expected due to obstacles in the indoor environment.
- We integrate two different explainable AI models into our deep learning model for providing local/global explanations of the trained model and generated predictions and interpreting the outputs of the model. Thus, our model can be used to identify which access points are more effective on each grid. This provides an explanation of how grids with very similar fingerprints are predicted correctly by the model. Because it presents the attributes (access points) and the reference values of the attributes that the model has learned for each grid. In this case, it also makes it easier for us to analyze which situations are effective in the mispositioning of a new sample.
- We present a comprehensive experimental analysis of different deep learning architectures used in indoor location estimation using three different WiFi-based indoor datasets. We demonstrate that our model works efficiently for environments where the number of access point devices is greater than the number of grids and for areas where the number of grids is greater than the number of access point devices.

The remainder of the paper is organized as follows. Related works are presented in Section 2. In Section 3, the proposed XAI-empowered deep learning system is presented. The experimental setup and experimental results are given in Section 4. Finally, the conclusions are discussed in Section 5.

2. Related works

Many indoor location detection approaches based on different approaches have been proposed in the literature [2,36]. In related studies, different types of signals have been used besides the use of images. Earth's magnetic field values, Bluetooth, RFID, and Infrared signals are some of the signals used for this purpose [36]. However, WiFi is the key technology to offer a ubiquitous indoor localization system with low additional hardware costs. The collection and processing of image data require the use of devices with high processing power and storage capacity compared to the use of electromagnetic signals. Earth's magnetic field signals produce very similar values in close fields and are subject to an effect called the hard iron effect [37]. When the magnetic field values of the Earth are collected, results that are pretty different from the original signal values can be encountered due to the presence of substances such as copper, nickel, and cobalt in the environment. The signal coverage of technologies such as RFID and infrared is low and causes the use of many hardware. On the other hand, WiFi technology reduces hardware costs. WiFi access points are used in the Internet infrastructure of almost every building. For this reason, WiFi signals have been frequently used as the primary technology in indoor localization approaches.

Indoor localization is a widely accepted subject of study with many sub-problems. Studies in the literature are presented in Table 1. After reviewing the studies in the literature on indoor positioning, it was found that there are numerous distinct approaches for the sub-problems in indoor positioning. These studies encompass selecting between deterministic and probabilistic approaches [21], coping with difficulties arising from changes in hardware infrastructure in dynamic environments [22], offering data augmentation techniques due to distortions occurring in signal collection [23], relying on AP selection to overcome signal distortions [24], using technologies other than WiFi [26,30], and developing algorithms for existing machine learning, artificial neural networks, and statistical techniques [27,28,31]. This study presents a Hybrid Deep Learning Architecture based on fingerprinting using WiFi technology. Therefore, Table 1 also presents deep learning studies based on fingerprinting using WiFi technology [10,14,17–20,35].

Orujov et al. [21] have developed an indoor localization approach based on Bluetooth Low Energy (BLE). In this approach, location is determined based on the techniques of Proximity Localization, Centroid Localization, Weighted Centroid Localization, Weight-Compensated Weighted Centroid Localization Based on RSSI, Fingerprinting, and Trilateration Localization. A fuzzy logic-based selection determines which technique to use for location determination in any given area, considering factors such as room width and signal strength. The highest accuracy was achieved with Fingerprinting. KNN and NN were used as classifiers for Fingerprinting-based localization. Li et al. [22] proposed a fingerprinting-based localization approach for environments where hardware infrastructure changes are possible. They presented a study that could work in areas with dynamic hardware infrastructure. They suggested using Long Short-Term Indoor Positioning (LSTP) as a solution to address both the challenge of working with heterogeneous feature spaces and the issue of dealing with continuous environmental dynamics at different time scales simultaneously. Sulaiman et al. [23] focused on the challenges encountered in the creation of a signal map in their fingerprinting-based study. To overcome the difficulties of dealing with signal readings during offline scans, they expanded the signal map using biharmonic spline interpolation. The authors used a feedforward backpropagation (FFBP) neural network and generalized regression neural network (GRNN) to perform location tracking in the online phase of their study. They trained their models with a semi-interpolated dataset, where the desired output was determined by the reference point's (X, Y) coordinates. They utilized two types of data as their inputs: the first being the RSSI values collected from 17 access points (APs), with three of them supporting both 2.4 and 5 GHz bands, and the second input consisted of a specific set of APs that produced acceptable RSSI levels and their respective coordinates. Chen et al. [24] propose a new algorithm called restricted weighted k-nearest neighbor (RWKNN) for determining the location of a mobile user using the fingerprinting technique. They aim to overcome the challenges of RSS instability and spatial ambiguity by modifying the traditional weighted k-nearest neighbor algorithm. The proposed algorithm considers indoor mobility constraints and uses searching rectangles and trajectory

Table 1
Different indoor localization studies in the literature.

Study	Techology(ies)	Method	Dataset (Conditions)	Result
Orujov et al. [21]	BLE Beacons	Fuzzy logic for selection KNN, NN	Office rooms and corridors	0.67 m LE
Li et al. [22]	WiFi BS	Knowledge Transfer	Office: 1460 m^2, 210 grid points. Library: 308.4 m^2, 48 grid points	0.84 m MLE 2.18 m ME
Sulaiman et al. [23]	WiFi APs	FFBP, GRNN	37×32 m^2 area, 17 APs	0.48 m LE
Chen et al. [24]	WiFi APs	RWKNN	Tampere dataset [25]	6.5 m average error
Kumar and Rajawat [26]	GSM, WiFi APs	Dictionary Learning Hidden Markov Model	Map1: grids around tables Map2: grids separated by 1 m Map3: grids separated by 25 cm Map4: grids separated by 50 cm	50% reduction in MLE.
Guo et al. [27]	USRP, RSS	MUCUS SWIM	9.8×6.3 m^2, 18 grids	MSE between 1 m and 4.5 m
Li et al. [28]	WiFi APs	TWKNN	Building: 3000 m^2, 322 RP APs: 97 UJIIndoorLoc [29]	2.93 MLE
Huo et al. [30]	FILS15.4	LQI	7 rooms, and RPs	Higher than 97% accuracy
Xue et al. [31]	WiFi APs	BPNN	53 RPs, 1.2 m intervals 47 TPs, 0.6 m intervals 60 RPs, 1.2 m intervals	0.87 m MLE
Mendoza-Silva et al. [32]	WiFi APs	Support Vector Regression	Library dataset [33] Mannheim dataset [34]	Accuracy below 4 m.
Chen et al. [20]	WiFi APs	LF, DLSTM	Lab: 35.3 m × 16.0 m. Office: 55.0 m × 50.0 m. 353 RP, 20 APs.	under 1.48 and 1.75 m MLE.
Hsieh et al. [35]	WiFi APs	LSTM	WiFi Fingerprinting Dataset [33]	99.7% floor accuracy
Song et al. [17]	WiFi APs	SAE, CNN	UJIIndoorLoc dataset [29] Tampere dataset [25] UTSIndoorLoc dataset [17]	96.03% floor accuracy 94.22% floor accuracy 94.57% floor accuracy
Khatab et al. [10]	WiFi APs	SAE, DELM (ADELM)	Lab: 40.4 m × 28.8 m, 15 APs, 19 RPs.	94.75%
Kim et al. [19]	WiFi APs	SAE, DNN	UJIIndoorLoc dataset [29]	99.8% building hit rate 91.27% floor hit rate
Hernandez et al. [18]	WiFi APs	CNN	3600 m^2 campus, 113 APs, 67 RPs.	3.5 MLE
Jia et al. [14]	WiFi APs	PGSE, LSTM	UJIIndoorLoc dataset [29] Library 1000 m^2 [14]	4 m MLE

limitations to reduce spatial uncertainty. Additionally, a confidence level is introduced to mitigate the effects of RSS instability on the iteration-based approach. Kumar and Rajawat [26] have proposed a positioning algorithm using signals obtained from GSM and WiFi access points. The algorithm used for learning a dictionary has been modified to prevent similarity between neighboring areas and to assign appropriate weights to the sparse coefficients. Moreover, they have incorporated a tracking algorithm based on hidden Markov models that take into account the recent history to estimate the user's location. To test their proposed approach, they used four maps containing grids of different shapes. They tested a wide range of scenarios and achieved a 50% reduction in the average error. Guo et al. [27] focused on group fingerprinting in their study. They proposed a fusion algorithm called MUCUS, which combines multiple classifiers and samples to improve localization accuracy. In addition, they introduced windowing and sliding techniques in their approach, referred to as SWIM, to enhance the efficiency of localization. They minimized the entropy of either multiple classifiers or multiple samples to obtain a more precise estimate of the user's location. Li et al. [28] proposed a new approach called TILoc to achieve effective location determination while avoiding the effects of RSSI signals in wireless networks. In this approach, they filtered out unstable access points (APs) and used online RSSI fingerprints. The authors used a part of robust APs to construct an RP torus and considered the RPs in the intersection of RP tori as the nearest RPs. They trained offline and online fingerprints using robust principal component analysis (RPCA) to reduce sparse spikes noise. Additionally, they took into account the AP's effect when positioning a mobile target. For the positioning step, they employed a weighted nearest neighbor strategy on target estimation (TWKNN) that assigns AP weight on target estimation. Huo et al. [30] presented a technique to optimize parameters in an indoor localization system that uses fingerprints based on IEEE802.15.4 (FILS15.4) to ensure the correct identification of fingerprints. Their approach involves an iterative process to modify fingerprint values to improve the system's room detection accuracy based on a newly defined score function. Additionally, the number of fingerprints for a particular room is automatically increased if the accuracy drops below a certain level. Xue et al. [31] introduced a new approach called high-adaptability indoor localization (HAIL) in their study, which combines the advantages of relative RSS values and absolute RSS values to improve the robustness and accuracy of the system. They use a backpropagation neural network (BPNN) in the HAIL approach to determine the similarity of fingerprints based on absolute RSS values, resulting in an average localization error (MLE) of 0.87 m. In their study, Mendoza-Silva [32] provided recommendations for selecting locations to collect WiFi samples and proposed a new model to predict received signal strength. The proposed model generates vectors that describe any obstructions between an access point and the collected samples. The distance between the access point and sample positions, along with the collected data, are also used to train a Support Vector Regression in the associated research.

Another feature sought in WiFi-based indoor localization systems based on fingerprinting technique is to offer a system with high performance by using as few access points as possible. Offering a highly accurate positioning system from a small number of access points inherently offers an overcomplete network structure [38]. To increase positioning accuracy in such networks, deep learning architectures

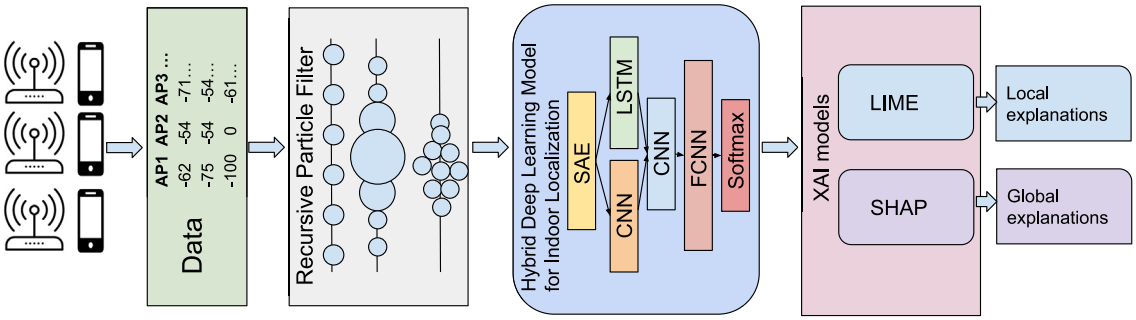


Fig. 1. The XAI-powered hybrid deep architecture for indoor localization.

form the basis of many studies with their powerful computing infrastructures. Chen et al. [20] suggested a local feature-based deep long short-term memory (LF-DLSTM) approach in their study. They collected data from two different indoor environments, a research lab, and an office. Accordingly, they collected signals from 353 uniformly placed reference points using a total of 20 access points. In the proposed method, they extracted local features from the raw data they collected, and the obtained raw features were classified using Deep LSTM - DLSTM architecture. Hsieh et al. [35] aimed to evaluate the effectiveness of Recurrent Neural Network (RNN) in indoor positioning systems and compared RNN and LSTM architectures. For this purpose, they used the Long Term WiFi Fingerprinting dataset and 46 800 of the data in the dataset for testing and 16 704 for training purposes. The data in the dataset is labeled according to the x and y coordinate information. Song and et al. [17] used three different datasets in their study. They normalized the existing data in the datasets and performed dimension reduction using stacked autoencoders. Thus, they aimed to preserve the necessary features of the data. In their work, the authors presented a CNN-based deep learning architecture and tested the performance of different optimizers. Khatab et al. [10] presented a deep extreme learning machine with a feature extraction using an autoencoder - ADELM. In order to test their work, the authors created a dataset from a laboratory environment covering an area of 40.4 m× 28.8 m. There are 19 access points and a total of 10 000 data in the created dataset. Kim et al. [19] presented a scalable deep neural network architecture for multi-building and multi-floor indoor localization in their study. For this purpose, they used the UJIIndoorLoc dataset, which contains many building and floor information. 3 floors and 5 different buildings were classified by using stacked autoencoders for feature extraction. The authors obtained 99.82% for the building hit rate and 91.27% for the floor hit rate. Hernandez et al. [18] presented a CNN-based approach, which they called WiFiNet in their work. The proposed CNN approach includes 13 convolutional layers, a Batch Normalization (BN) layer and a ReLU layer (Rectified Linear Unit) every three Conv+BN layers. The method was tested using SVM and subspaceKNN using ResNet18 and AlexNet for feature extraction. Accordingly, the highest accuracy was obtained by reaching 3.5 MLE for existing positions with WiFiNet. Jia et al. [14] proposed a WiFi fingerprint-based localization algorithm based on Long Short-Term Memory Network (LSTM). The authors used a sample expansion algorithm (PGSE) based on principal component analysis and sparse-sample Gaussian process regression. In particular, when the number of collected reference points is limited, principal component analysis is used to select access points, and Gaussian process regression is used to extract the reference point coordinates and corresponding received signal strength values in the training samples.

3. Proposed XAI-based deep architecture

In this section, we present the proposed XAI-powered deep learning architecture. First, we give explanations about the pre-processing process. Second, we present the architecture of our hybrid deep learning model for indoor localization. Then, we describe the XAI models that we use for interpreting and explaining the outputs of our deep learning model. Fig. 1 shows the overall system architecture for indoor localization. Each phase in the given architecture is explained in detail in the sub-sections.

3.1. Pre-processing: Particle filter

To remove the unwanted effects from the signal values in the signal data and to explore a space that converges to the original values, we use a particle filter to data as a pre-processing phase. A particle filter is a Bayesian theory-based filter based on approximations that find the most likely candidate to implement the system. Particles in the particle filter are assigned weight values that represent possible solutions. The weight values indicate how strongly the solution represented by the particles affects the posterior density function. The posterior distribution is calculated by updating the particle weights and resampling, considering the measurements performed. The particle filter includes two recursive steps, a prediction step, and an update step. At each step, the estimation process is assigned weights that indicate the significance of particles representing multiple copies of the variable of interest [39]. The estimation step is performed using the weighted sum of all particles. The particle filter algorithm mimics the iterative steps found in nature. After each operation, each particle value is updated by adding random noise that is re-evaluated according to observations [9]. In a recursive operation, particles with lower weights are removed from the system, while particles with higher weights remain in the system until convergence is maximized.

3.2. Proposed hybrid deep learning model

Fig. 2 shows the overall architecture of the proposed deep learning model for indoor localization. The proposed architecture is a hybrid deep learning method that comprises a sparse autoencoder for feature extraction and filtering process, two convolutional layers and maximum pooling layers to capture local patterns, one LSTM layer to consider long-term dependencies, and the two-layer fully connected neural network and softmax layer for classification. In this section, these phases are explained in detail, respectively.

Sparse autoencoders (SAEs) provide representative learning using neural network-based models that simply consist of an encoder, a bottleneck, and a decoder [40]. Autoencoders provide a bottleneck in the network that provides focusing on extracting the compressed information representation of the original input, capturing the more meaningful and distinctive part of the data, and eliminating potential noise, redundant and irrelevant features within the data. Therefore, autoencoders can also act as a filter that takes into account the correlation between features, especially for signal data [41]. Sparse autoencoders generally employ a greater number of hidden nodes compared to the input layer. In particular, it is an over-complete autoencoder example that is used to identify significant features within data that possess a

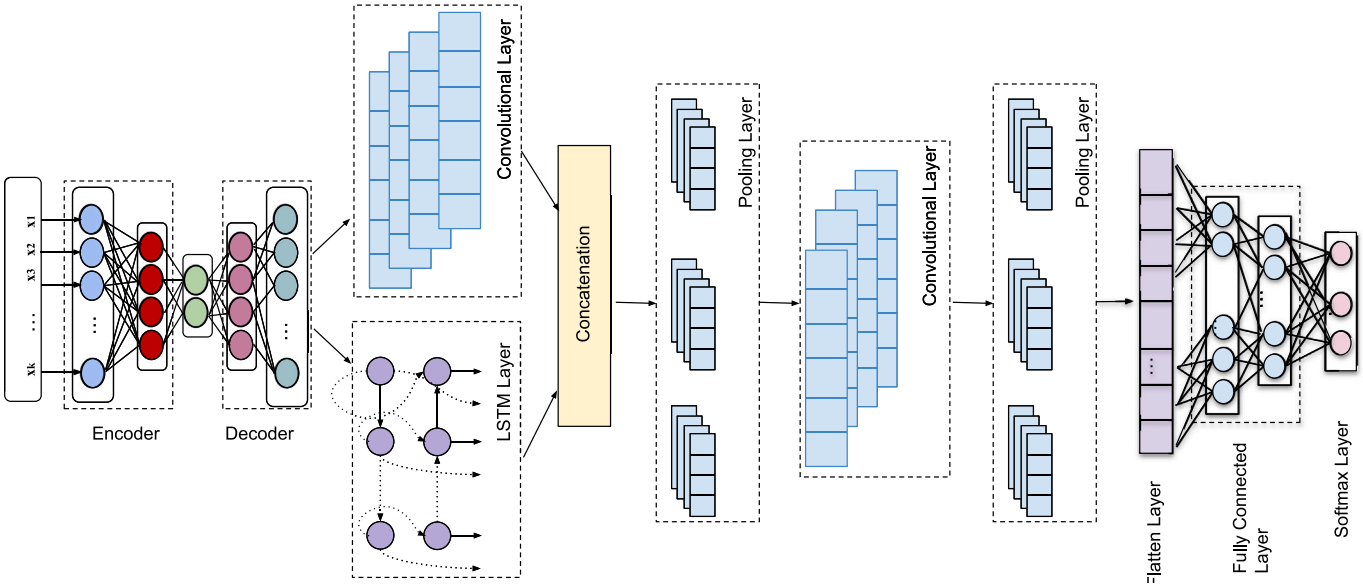


Fig. 2. The architecture of proposed hybrid deep learning model for indoor localization.

limited number of features. By applying a sparsity constraint to the hidden units, the autoencoder can reveal noteworthy patterns in the data, even with a large number of hidden units. Since neural networks cannot set a flexible number of nodes in hidden layers, the autoencoder achieves this by penalizing the activation of certain neurons in hidden layers. This is done by introducing a penalty term to the loss function, encouraging only a few neurons to be active within a layer. There are two primary methods to implement the sparsity penalty: L_1 regularization and KL-divergence.

A SAE generates a compressed latent representation of an input vector $\boldsymbol{x} \in R^{D_x}$, using the encoder and decoder units without the label information. Given an input vector **x**, the activation of all hidden neurons are calculated as

$$h^1(\boldsymbol{x}) = f^1(\boldsymbol{W}^1\boldsymbol{x} + \boldsymbol{b}^1) \tag{1}$$

where $f^1 : R^{D^1} \rightarrow R^{D^1}$ is a transfer function of the encoder for first layer (1), $\boldsymbol{W}^1 \in R^{D^1} \times R^{D_x}$ is a weight matrix, and $\boldsymbol{b}^1 \in R^{D^1}$ is a bias vector. The output vector $\hat{\boldsymbol{x}}$ is then calculated as

$$\hat{\boldsymbol{x}} = f^2(\boldsymbol{W}^2 h^1(\boldsymbol{x}) + \boldsymbol{b}^2) \tag{2}$$

where $f^2 : R^{D_x} \rightarrow R^{D_x}$ is the transfer function of the decoder for second layer (2), $\boldsymbol{W}^2 \in R^{D^x} \times R^{D_1}$ is a weight matrix, and $\boldsymbol{b}^2 \in R^{D_x}$ is a bias vector. To minimize the reconstruction error, the weight matrices W^1 and W^2 and the bias vectors b^1, b^2 are adjusted by the back propagation algorithm. The cost function J of SAE is given as follows:

$$J = \frac{1}{N}\sum_{i=1}^{N} \|\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i\| + \lambda * \frac{1}{2}\sum_{i=1}^{2} \|\boldsymbol{W}^i\|^2 + \beta * \sum_{k=1}^{D^1} KL(p \parallel \hat{p}_i) \tag{3}$$

The cost function of a sparse autoencoder is an adjusted mean squared error function. The first term is the average squared error between all input data and output data for all N input data. The second term comprises a weight decay parameter λ that is the coefficient to tune the weights decay between the hidden and output units, and the L_2 regularization term that provides a regularization on the weights to the cost function to control the decrease of the output values depending on the values of the weights when training SAE. The last term consists of sparsity penalty that is the Kullback–Leibler (KL) divergence between the desired sparsity level and the actual average activation of each hidden unit. The hyperparameter β is used for controlling the strength of the penalty. Adding sparsity to an autoencoder is possible by adding a regularizer to the cost function that imposes a constraint on the sparsity of output from the hidden layer. Sparseness is achieved by using a sparsity regularization term that takes a large value to maintain low mean activation values when the average activation value $\hat{\rho}_i$ of a neuron i is not close to the desired value ρ. The equation of the Kullback–Leibler divergence is given as follows:

$$KL(\rho \parallel \hat{\rho}_i) = \rho * log(\frac{\rho}{\hat{\rho}_i}) + (1-\rho) * log(\frac{(1-\rho)}{(1-\hat{\rho}_i)}) \tag{4}$$

This penalty function outputs zero when $\hat{\rho}_i$ and ρ are equal, and otherwise outputs larger value that increases monotonically as they diverge from each other. In the process of optimizing the cost function, this term is reduced so that $\hat{\rho}_i$ and ρ converge. A large sparsity penalty encourages the activation of only a small subset of hidden units, supporting the discovery of higher-level features. However, the error rate may increase as the expressive capacity of the model is reduced during the input reconstruction process. On the other hand, a small sparsity penalty allows for more hidden unit activation, enhancing accurate reconstruction but potentially making the model more sensitive to noise and less interpretable.

The training process of an SAE takes place by minimizing the cost function J with a certain number of iterations. After reducing the cost function to a very small desired value, automatic feature extraction is provided with SAE. In the next phase, the output of SAE is passed to a $1D$ convolutional layer. $1D$ convolution layer is generally used in Natural Language Processing applications for extracting the most influential multi-terms and capturing local features. Similarly, the proposed method applies a convolutional layer to extract the most discriminative local patterns for the features obtained from the feature extraction process with SAE [42]. The convolutional layer applies a set of k filters the sub-matrices of the output matrix $\hat{\boldsymbol{X}}$ of SAE. Each filter $F \in R$ of size $l \times w$, where l and w refer to the height and width of the convolution filter, is applied to a window of l words to generate a new feature v_i from a window of vectors $\hat{\boldsymbol{X}}_{i:i+l-1}$ as follows:

$$v_i = f(F * \hat{\boldsymbol{X}}_{i:i+l-1} + b) \tag{5}$$

where $b \in R$, f and $\hat{\boldsymbol{X}}_{i:i+l-1}$ is the bias, the nonlinear activation function Rectified Linear Unit ($ReLU$) and the concatenation of $\hat{\boldsymbol{X}}_i, \ldots, \hat{\boldsymbol{X}}_{i+l-1}$, respectively. The filter scans each possible window of the matrix $\hat{\boldsymbol{X}}$ and performs convolution operations to produce a feature map m where m corresponds to $[v_0, v_1, \ldots, v_{n-l+1}]$. By applying k convolution filters, k different feature maps are produced. For the last component of the convolutional layers, ReLU function is used as an activation

function to increase the non-linearity in the outputs. ReLU function sets all negative values in the vectors to zero keeping all positive values. The zero padding is applied to preserve the original size of all sequences when applying a convolutional filter. Afterwards, the feature maps are combined $M = \{m_1, m_2, \ldots, m_k\}$ as the output of the convolutional and nonlinear layer.

To reduce the dimension of feature maps, it is common practice to use a pooling layer. The proposed system utilizes a pooling layer that applies max-pooling operation over the feature map m and extracts the maximum value $\hat{m} = max(m)$ as the final feature. This pooling process provides discovers the most dominating feature of each filter. After capturing k features from the feature map, the pooling results are combined $\hat{m} = \hat{m_1}, \hat{m_2}, \ldots, \hat{m_k}$ as the output of the CNN layer.

Meanwhile, the proposed method applies the LSTM layer in parallel with the CNN layer to capture long-term dependencies between SAE-derived features. This layer aim to capture high-order data correlations and patterns considering previous outputs as inputs. LSTM layer is fed by the input matrix $\hat{\boldsymbol{X}}$. It treats each feature of the sample as a separate input occurring at time t. At time-step t, the memory c_t and the hidden state h_t are updated with the following equations:

$$\begin{aligned} f_t &= \sigma(\hat{\boldsymbol{X}}_t W^f + h_{t-1} U_f + b_f) \\ i_t &= \sigma(\hat{\boldsymbol{X}}_t W^i + h_{t-1} U_i + b_i) \\ o_t &= \sigma(\hat{\boldsymbol{X}}_t W^o + h_{t-1} U_o + b_o) \\ C_t &= tanh(\hat{\boldsymbol{X}}_t W^i + h_{t-1} U_c) \\ c_t &= \sigma(f_t * c_{t-1} + i_t * C_t) \\ h_t &= tanh(C_t) * o_t \end{aligned} \tag{6}$$

where W_f, W_i, and W_o are the weight matrices, and b_i, b_f and b_o are biases. U_f, U_i, and U_o are weight matrices that provide recurrent connection between the previous hidden layer and the current hidden layer. C is the new candidate state that is created by *tanh* layer busing the current input and the previous hidden state, and c is the value of the memory unit that is computed using the previous memory, multiplied by the forget gate, and the newly generated candidate state, multiplied by the input gate. σ and *tanh* refer to the sigmoid function and the hyperbolic tangent function, respectively. The first step is to decide which information should be retained and which should be removed from the cell. In forget gate layer, the current input ϕ_t and the previous hidden state h_{t-1} are passed through the sigmoid function. The function outputs a number between 0 and 1. If the value is close to 1, the information is kept completely. If the value is close to 0, the information is completely forgotten. Input gate passes h_{t-1} and $\hat{\boldsymbol{X}}_t$ into a sigmoid function to determine which information to be updated. The new candidate cell C_t is determined using a *tanh* function. Thereafter, the output value is determined by the cell state c_t. The input from the previous cell state c_{t-1} is multiplied by the forget gate output f_t. This output is added with the input gate output i_t to update the new candidate cell state C_t. The output gate passes h_{t-1} and current input $\hat{\boldsymbol{X}}_t$ to the sigmoid function. To create the present hidden state, this output is multiplied by the output of the *tanh* function of C_t. The current state c_t and the present hidden state h_t are the final outputs of the LSTM network. The weight matrices of the forget, input, and output of the LSTM are represented by U_f, U_i, and U_o, respectively.

In the next step, the feature maps obtained from both convolutional and LSTM layers are concatenated to form a single feature matrix as follows:

$$L = [\hat{M} \oplus H] \tag{7}$$

To extract discriminative spatial features from the merged feature map, the same convolutional layer and maximum pooling layer described above are applied. The output of pooling layer is then passed to a flatten layer to convert the multi-dimensional maps into a single dimensional vectors. In the next phase, two-layer fully connected feed-forward neural networks are performed with ReLU activation function. The layers consist of 1024 and 512 hidden units, respectively. Last hidden layer is followed by a softmax layer that returns the probability score of each class. To train the network using the back-propagation algorithm, we use Adam stochastic optimizer, which is an algorithm for first-order gradient-based optimization of stochastic objective functions.

3.3. Explainable artificial intelligence models

Explainable Artificial Intelligence models aim to reveal the decision-making mechanisms of learning models and thus provide transparency and interpretability to these models [43]. In particular, deep learning architectures developed with the aim of achieving high accuracy in the learning and prediction phases for many problems are getting more and more complex, and therefore, it becomes impossible for both users and experts in order to analyze the decision stages of these models. Similarly, for indoor localization problem, it is important to understand and interpret which access points cause the model to predict the location correctly or incorrectly, or the contribution of the WiFi signal values received from these points to the prediction. The proposed framework uses two main XAI techniques such as Interpretable Model-Agnostic Explanations (LIME), and SHapley Additive exPlanations (SHAP) to generate both local and global explanations of the decisions of our hybrid deep learning method for all existing and new coming indoor data.

LIME uses local and interpretable model to explain each specific prediction. It does not give a general explanation of the common decision mechanisms that the model learns by considering all the data, but rather explains according to which local features a particular observation is categorized [44,45]. More formally, for an observation x, the explanations generated by LIME can be obtained by the following:

$$\phi(x) = argmin_{g \in G} \delta(f, g, \varphi_x + \Omega(g)) \tag{8}$$

where $g \in G$ is the explanation model and G is a set of interpretable models (e.g. decision tree, convolutional neural network). $\Omega(g)$ corresponds to the complexity of the explanation of all $g \in G$. φ_x refers to a proximity weight between the prediction of the explanation model and the original model. The aim is to minimize the loss function δ that measures the proximity between model g and the original model f that is used for classification.

Unlike LIME, instead of just revealing local decisions, SHAP calculates Shapley values that denote the average of the marginal contributions of all possible features, taking into account the contribution of each feature to the final decision [46]. In this way, it provides general interpretability of model decisions. The Shapley values for a given model f are calculated as follows:

$$\zeta_i(f, x) = \sum_{s \subseteq F} \frac{|s|!(M - |s| - 1)!}{M!} [f_x(s \cup i) - f_x(s)] \tag{9}$$

where $\zeta_i(f, x)$ is the Shapley value of feature i of a given observation x for model f, s denotes the subset of features, and F is the set of all features available in the original set.

4. Experimental setup and results

In this section, first, we give the details of datasets and the experimental setup for our proposed method and evaluation methods. Second, we present the experimental results of the proposed XAI-empowered deep model by comparing it to different deep learning models. The datasets used in the experiment, the evaluation methods, the experimental setup, and performance comparisons are described below, respectively.

Table 2
The details of datasets.

Features (Number of)	Halic	RFKON	UJIIndoorloc
Training samples for each fold	8896	4320	4199
Test samples for each fold	2224	1080	1050
Access points	12	37	520
Reference points	123	54	256
Floors	N/A	N/A	4

4.1. Datasets

The experiments have been conducted on publicly available three different indoor localization datasets using the fingerprinting method. This study was carried out using datasets created by the fingerprinting method, which is one of the deterministic-based scene analysis techniques. The relevant method consists of two stages; offline and online. Accordingly, the fingerprinting method is based on the mapping of the signal in the offline phase of the area where the location is desired to be determined and positioning by comparing the signals received from the mobile user in the online phase. The most important step in fingerprint methods is to present an algorithmic approach that enables the successful interpretation and learning of the created signal map.

We give the details of the datasets in Table 2. HALIC, RFKON, and UJIIndoorLoC datasets which are commonly used in the literature [9,29,47] were examined and arranged in accordance with the purpose of the study, and sub-datasets HALIC, RFKON, UJIIndoorLoc datasets containing only WIFI signals were created or used. The 5-fold cross-validation is used for creating training and test data from the training sets belonging to the datasets.

In the pre-processing stage, the grids in the datasets where the signal could not be collected were examined. If the points where the signal cannot be collected are expressed using positive integers, the relevant value is changed to "0". Then, a particle filter was used to estimate the original signal values by eliminating the effects caused by collecting the data from indoor areas on the top.

4.1.1. HALIC dataset

HALIC dataset has been collected in a university environment by the author in this paper [9]. A sub-dataset was created by selecting the WiFi signals in the HALIC dataset, and all experiments were performed on the relevant dataset. HALIC dataset is created using a single floor from HALIC University Sutluce Campus. There are 4 offices of different sizes, a corridor area, a kitchen, and two bathrooms belonging to academic members on the floor where data collection is carried out. There are 12 access points in the building. In the division of the grid areas where the WiFi RSSI values are collected, the places where people are likely to be found are based. Therefore, no reference point has been assigned to places where things can be found (bookshelves, refrigerators, etc.) and where people cannot be found. Empty areas that are not occupied by furniture and mobile people can be found are divided into grid areas of 4 m × 4 m, and their midpoints are chosen as reference points. Classes containing offices, kitchen, hallway, and bathroom areas, the total regions they cover, the number of reference numbers, the number of access points they have, and the brands/models of the relevant access points are presented in Table 3.

4.1.2. RFKON dataset

The RFKON dataset has been created to be used in indoor positioning studies and contains different sub-datasets [47]. In this study, the which contains only WiFi signals were used. The RFKON dataset was created using two of the Eskisehir Technopark buildings. The dataset, which includes 37 access points in total, contains 54 reference points on the first floor. Reference points are assigned to the midpoint of the 1.2 m × 1.2 m grid zones. In this study, the location of the mobile user is determined by using the RFKON dataset, which contains a total of 5400 data.

Table 3
The details of HALIC dataset.

Clas number	Reference number range	Access point number (IDs)	Location determined area
1 (open office)	1–47	4 (0,1,2,3)	188 m^2
2 (open office)	48	–	48 m^2
3 (open office)	49	1 (4)	49 m^2
4 (open office)	50–69	3 (5,6,7)	76 m^2
5 (open office)	70–104	3 (8,9,10)	136 m^2
6 (hallway)	105–107	–	8 m^2
7 (kitchen)	108	–	4 m^2
8 (hallway)	109–111	–	8 m^2
9 (hallway)	112–123	1 (11)	44 m^2

4.1.3. UJIIndoorLoc dataset

It is a dataset created with WiFi RSSI values collected from 520 access points from many buildings and many floors. It contains only RSSI information of WiFi signals. In the relevant dataset, there are Longitude, Latitude, Floor, BuildingID, SpaceID, RelativePosition, UserID, PhoneID and Timestamp information presented with the RSSI values of the access points. In this study, the sub-dataset was created by selecting the data with a BuildingID of 0. Sub-dataset contains a total of 256 reference points. In UJIIndoorLoc, the SpaceIDs of the data are assigned considering the office, corridor, and classroom information. The selected building has 4 floors. Reference points were determined independently of floors. Separate id assignments were made for locations with the same space ID but different floor numbers. In the new created dataset, different labels are given to each floor's office, corridor, and space zones. Thus, a new sub-dataset was created, allowing the determination of each region on each floor in building number 0. The dataset contains only assigned labels and RSSI values from 520 access points.

4.2. Experimental setup and evaluation methods

Five different deep learning models and two machine learning methods are selected as baseline methods for evaluating the performance of the proposed method. We use Random Forest (RF) and Extreme Gradient Boosting (XGBoost) as machine learning methods. DNN, CNN, LSTM, and CNN-LSTM which are frequently used methods for indoor localization in the literature [20,48] were implemented. For the CNN model, we performed multiple versions of CNN by varying the number of convolution layers from 1 to 3. The best results for all datasets were obtained by CNN which consists of one convolutional layer and one pooling layer. We tested the number of filters in the range of [16, 32, 64], and the kernel size in the range of [3, 5, 7]. Thereafter, we empirically set the number of filters and the kernel size as 32 and 3, respectively. For the LSTM network, we tested the number of hidden units in the range of [16, 32, 64]. Considering the accuracy performance in experimental tests, we set the number of memory units to 32. CNN-LSTM includes one convolutional layer, one max-pooling layer, and then a LSTM layer, respectively. For the CNN-LSTM network, we used the same settings as aforementioned.

Similarly, we used the same parameter settings for convolutional and LSTM layers in our model as given in Table 4. For the SAE, the number of hidden units was fixed to 60, and the sparsity penalty proportion was set to 0.3. The value of the sparsity penalty proportion is adjusted depending on the dataset, the complexity of the task, and the desired sparsity level. Finding the appropriate value for this parameter usually requires experimentation and adjustment by grid search, depending on domain knowledge and prior experience. The values used for the SAE are set by applying grid search. The activation functions for the encoder and decoder are selected as satlin and purelin, respectively. The number of epochs was set to 1300. The number of hidden layer for dense fully connected network is set to 2. The size of hidden units is set to 1024 and 512, respectively. The softmax activation function is

Table 4
Experimental setup for the proposed model.

SAE		Convolutional (CL) and LSTM Layer (LL)		Dense and output layers	
Number of epochs	1300	CL: Number of kernels	32	Number of neurons	1000, 500
Sparsity regularization	2.5	CL:Kernel size	3	Activation Function	ReLU
Sparsity penalty proportion	0.3	CL:Activation function	ReLU	Output Activation Function	Softmax
Number of neurons	60	CL:Pooling	MaxPooling	Optimizer	Adam
Encode transfer function	Satlin	LL: Number of hidden units	32	Learning Rate	0.001
Decode transfer function	Purelin			Number of neurons for output	123, 54, 256

Table 5
Output shape of each layer in the proposed method.

Layers	Output shape
SAE input layer	(12), (37), (520)
SAE bottleneck	(60)
SAE output layer	(12), (37), (520)
Convolutional layer	(60,32)
LSTM layer	(60,32)
Concatenate layer	(60,64)
MaxPooling layer	(30,64)
1 convolutional layer	(28,32)
MaxPooling layer	(14,32)
Flatten layer	(448)
Dense layer	(1000)
Dense layer	(500)
Output layer	(123), (54), (256)

selected as the multi-label classification. The training batch size and epoch are set to 32 and 200, respectively. To prevent overfitting, early stopping with monitoring validation loss in max mode with patience of 5 is used in the training process.

Output shape of each layer in the proposed method is given in Table 5. For each dataset, the output shape of input layer and output layer is different. The output shapes are 12, 37 and 520 for HALIC, RFKON, UJIIndoorLoc, respectively. Similarly, the output shapes of output layer are 123, 54, 256 for HALIC, RFKON, UJIIndoorLoc, respectively. We use the representation obtained from bottleneck of SAE for dimensionality scaling. Thus, the output shape of convolutional layer after SAE output layer is obtained as (60,32) with 32 kernels.

Experiments were implemented on a 64-bit operating system running on the Intel Core i7-5930K CPU working 3.5 GHz processor and 24 GB memory.

4.3. Classification results for indoor locations

In this section, we present the accuracy (ACC), F-score, Precision (PREC), and Recall (REC) values belonging to the different methods for indoor location classification. We analyze the performance of our model and the other models on three different datasets.

We evaluate the results under related subsection for each dataset.

4.3.1. HALIC dataset

According to Table 6, it is observed that higher accuracy results are obtained with the filtering process for HALIC dataset when compared to results obtained from the raw version of dataset. The accuracy value of DNN increases from about 0.1790 to 0.8814. These results confirm that the particle filter is very effective to regulate the RSSI values.

The highest accuracy performance was obtained with RF method using filtered version of HALIC dataset. XGBoost achieves the second highest accuracy. Among the deep learning architectures applied directly to the data after the filtering step, the highest performance was obtained with DNN and the second highest performance with CNN-LSTM for HALIC dataset. However, it is seen that deep learning methods achieve 8 percent lower performance than RF and XGBoost. We assume that the most important reason for this is the limited number of features as there are only 12 APs in the HALIC dataset. It is concluded that a 1×12 space is not sufficient for convolutional and

Table 6
Performance comparison of the proposed method and baseline methods for HALIC dataset.

	Methods	ACC	F-SCORE	PREC	REC
Raw	RF	0.2109	0.2448	0.2124	0.2143
	XGBoost	0.2001	0.2339	0.2014	0.2036
	DNN	0.1790	0.2277	0.1801	0.1743
	CNN	0.1665	0.2179	0.1686	0.1593
	LSTM	0.1955	0.2461	0.1968	0.1941
	CNN-LSTM	0.1938	0.2393	0.1945	0.1898
Filtered	RF	0.9669	0.9676	0.9674	0.9664
	XGBoost	0.9562	0.9576	0.9569	0.9558
	DNN	0.8814	0.8912	0.8808	0.8763
	CNN	0.8676	0.8693	0.8640	0.8575
	LSTM	0.8702	0.8688	0.8677	0.8611
	CNN-LSTM	0.8766	0.8849	0.8757	0.8718
Filtered and SAE	RF	0.9701	0.9712	0.9704	0.9694
	XGBoost	0.9563	0.9568	0.9571	0.9553
	DNN	0.9688	0.9693	0.9846	0.9808
	CNN	0.9737	0.9751	0.9730	0.9727
	LSTM	0.9739	0.9749	0.9738	0.9731
	CNN-LSTM	0.9764	0.9768	0.9764	0.9757
Proposed method		**0.9852**	**0.9851**	**0.9858**	**0.9849**

similar operations that deep learning models use for deep feature engineering. When the sparse autoencoder is applied to the data after the filtering step, it has been determined that the accuracy performances of these deep learning architectures increase remarkably. With the usage of SAE with models, it is observed that the classification performances of the deep learning methods for HALIC dataset can improve by an average of at least 11%. There are two major reasons why SAE can significantly improve performance. Firstly, this may be because the input size has been increased from 12 to 60 using SAE. HALIC dataset contains 12 APs and 123 reference points. It is an example of an over-complete network where the size of the output is larger than the size of the input. Autoencoders generally provide high performance in over-complete networks that represent obtaining high precision from a small number of features. SAE provides dimension expansion for the dataset having limited features. Thus, a more suitable space is created for applying the operations of deep learning models. The second reason may be that SAE generates more meaningful representation by eliminating potential noise, redundant and irrelevant features within the data.

The results show that CNN-LSTM with SAE achieves better performances than the other models using SAE in terms of accuracy and f-score. However, in general, the results demonstrate that the deep models with SAE achieve very similar performances. The best results are achieved by the proposed method. The results verify that the deep feature fusion approach that uses enriched feature space obtained from both CNN and LSTM is more effective at capturing distinctive features for reference points with similar RSSI values in the same room or corridor.

4.3.2. RFKON dataset

The results given in Table 7 show the classification performance of the methods for RFKON dataset. Similarly, by applying the filtering process, higher accuracy results were obtained than all the basic methods. The accuracy value of DNN increases from about 0.6817 to 0.9690. The

Table 7
Performance comparison of the proposed method and baseline methods for RFKON dataset.

	Methods	ACC	F-SCORE	PREC	REC
Raw	RF	0.7768	0.7857	0.7711	0.7687
	XGBoost	0.8999	0.8982	0.8953	0.8933
	DNN	0.6817	0.6847	0.6800	0.6755
	CNN	0.7106	0.7151	0.7094	0.7042
	LSTM	0.7391	0.7404	0.7339	0.7302
	CNN-LSTM	0.7167	0.7162	0.7113	0.7068
Filtered	RF	0.9766	0.9755	0.9759	0.9747
	XGBoost	0.9755	0.9744	0.9740	0.9733
	DNN	0.9690	0.9670	0.9680	0.9659
	CNN	0.9731	0.9716	0.9701	0.9695
	LSTM	0.9612	0.9578	0.9571	0.9545
	CNN-LSTM	0.9688	0.9626	0.9654	0.9624
Filtered and SAE	RF	0.9759	0.9741	0.9757	0.9741
	XGBoost	0.9758	0.9742	0.9741	0.9730
	DNN	0.9722	0.9709	0.9708	0.9694
	CNN	0.9746	0.9756	0.9738	0.9736
	LSTM	0.9607	0.9611	0.9577	0.9563
	CNN-LSTM	0.9741	0.9738	0.9725	0.9721
Proposed method		**0.9842**	**0.9905**	**0.9984**	**0.9901**

best result is achieved by RF and XGBoost for the filtered version of the dataset. The performance of these methods is followed by that of CNN and CNN-LSTM.

SAE increases the performances of DNN, CNN and CNN-LSTM by approximately between 0.03% and 0.05%. Unlike the HALIC dataset, it is observed that the sparse autoencoder does not make a significant contribution to this dataset. Although RFKON presents an over-complete network, the difference between the number of access points and reference numbers are approximately balanced and lower than HALIC. For this dataset, the proposed method also outperforms the baseline methods. The proposed method increases the performance by approximately 1.52% in terms of F-score.

4.3.3. UJIIndoorLoc dataset

Table 8 presents the classification performance results for UJIIndoorLoc dataset. LSTM achieves higher performance than the other methods for the raw version of the dataset. The accuracy value of LSTM increases from about 0.6905 to 0.9308 when the model takes the filtered version of the dataset as input. Significant increases at different levels are also observed for other learning models. CNN with SAE model obtains higher performances in terms of accuracy, precision, recall, and f-score. The performance of this model is followed by CNN-LSTM with SAE model. It is seen that the contribution of SAE to performance is not notable. Therefore, the deep learning methods with SAE and without SAE obtain very similar accuracy results. However, for this dataset, it provides computationally contribution as it performs a dimension reduction by reducing the feature space of 520 to 60. Moreover, it is observed that all basic methods obtain almost the same accuracy values as each other. The proposed method outperforms all baseline methods by increasing the accuracy by approximately 2% over the filtered version of the dataset.

Our method learns the locations in the UJIIndoorLoc dataset regardless of the floor information. The method runs the proposed system in parallel to determine the floor number. It returns the specified reference point and floor number to the user in this way. Building 0 has 4 floors. Therefore, we use four neurons in softmax layer. Table 9 shows the floor classification result of the methods. Even with the original values of the dataset for floor classification, 99.60% accuracy is reached with the XGBoost algorithm. The accuracy scores obtained after the filtering process are found to be 100% with deep learning models.

Table 8
Performance comparison of the proposed method and deep learning methods for UJIIndoorLoc dataset.

	Methods	ACC	F-SCORE	PREC	REC
Raw	RF	0.5925	0.6084	0.5952	0.5641
	XGBoost	0.7369	0.7387	0.7352	0.7093
	DNN	0.4390	0.4507	0.4345	0.4100
	CNN	0.4622	0.4496	0.4603	0.4159
	LSTM	0.6905	0.6904	0.6918	0.6993
	CNN-LSTM	0.5881	0.5983	0.5858	0.5536
Filtered	RF	0.9148	0.9107	0.9093	0.8961
	XGBoost	0.9261	0.9225	0.9201	0.9102
	DNN	0.9167	0.9172	0.9116	0.8998
	CNN	0.9337	0.9303	0.9296	0.9176
	LSTM	0.9308	0.9263	0.9244	0.9144
	CNN-LSTM	0.9293	0.9269	0.9237	0.9127
Filtered and SAE	RF	0.9190	0.9150	0.9130	0.9022
	XGBoost	0.9178	0.9118	0.9107	0.8974
	DNN	0.9178	0.9159	0.9124	0.9004
	CNN	0.9339	0.9307	0.9301	0.9182
	LSTM	0.9318	0.9272	0.9254	0.9175
	CNN-LSTM	0.9327	0.9297	0.9266	0.9166
Proposed method		**0.9533**	**0.9446**	**0.9563**	**0.9417**

Table 9
Floor classification results of the proposed method and baseline methods for UJIIndoorLoc dataset.

	Methods	ACC	F-SCORE	PREC	REC
Raw	RF	0.9703	0.9700	0.9710	0.9705
	XGBoost	0.9960	0.9960	0.9959	0.9960
	DNN	0.9409	0.9417	0.9421	0.9416
	CNN	0.9857	0.9859	0.9860	0.9859
	LSTM	0.9970	0.9968	0.9970	0.9969
	CNN-LSTM	0.9970	0.9968	0.9970	0.9969
Filtered	RF	0.9990	0.9991	0.9991	0.9991
	XGBoost	0.9990	0.9991	0.9991	0.9991
	DNN	1	1	1	1
	CNN	1	1	1	1
	LSTM	1	1	1	1
	CNN-LSTM	1	1	1	1
Proposed method		**1**	**1**	**1**	**1**

4.3.4. Overall results

In Tables 6–8, it is observed that higher accuracy results are obtained with the filtering process for all datasets, when compared to results obtained from the raw version of datasets. The highest performance increase with the filtering process is observed in the HALIC dataset. The most important reason for such low performance from the raw version of this dataset is that the HALIC dataset has open office areas where mobility is very high. This human activity detracts the signals from the original values. Fig. 3 shows the effects of particle filter on RSSI values for HALIC dataset. Fig. 3 contains samples of the raw signal collected from room 5 and the particle-filtered results. Access point 9, access point 10 and access point 11 are in room 5. On the other hand, RSSI values received from access points 9, 10, and 11 in class 5 are equal to zero at some time intervals. Receiving values close to −100 or −100 consecutively from an access point and zero values immediately after these values indicate that there may exist some measurement errors or various effects on the signal values such as signal blocks or obstacles. In addition, while the RSSI values from AP 9 fluctuate mostly between −100 and −50, the values from AP 10 and AP 11 are more intensely close to −100. This may indicate the presence of an obstacle that weakens the signals obtained from AP 9. It was observed that the range of the signal values changed after the use of the particle filter, and the signal values equal to zero were regulated according to the high RSSI values obtained before. These results confirm that the particle filter is very effective to regulate the RSSI values that fluctuate over time, which cannot be obtained with

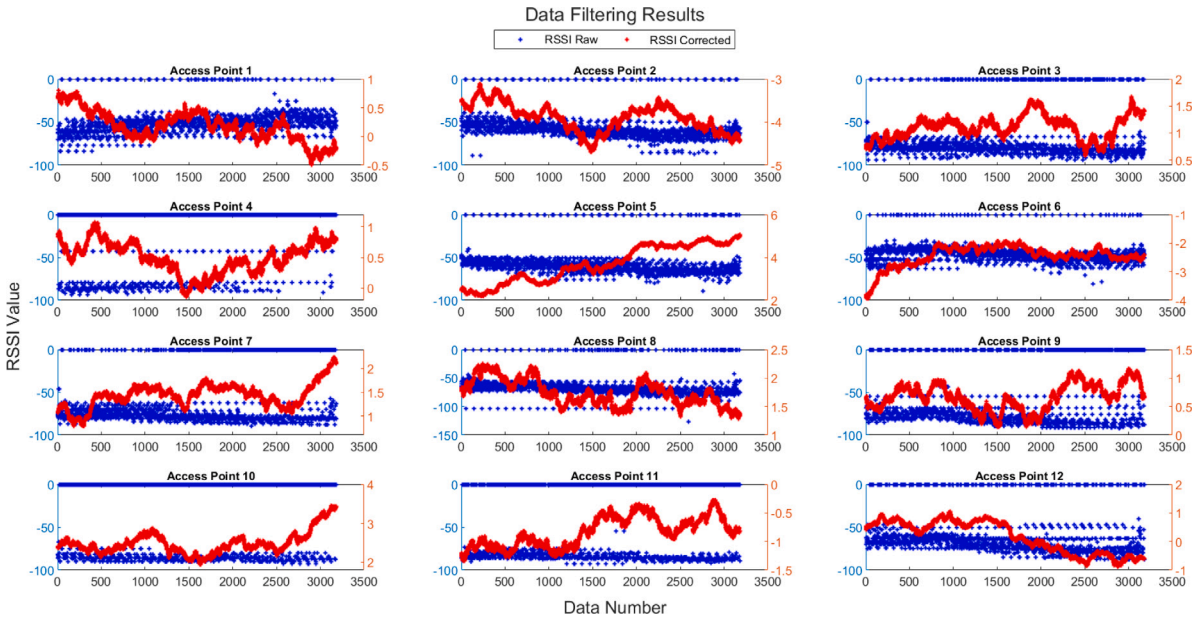


Fig. 3. The effects of particle filter on RSSI values for HALIC dataset.

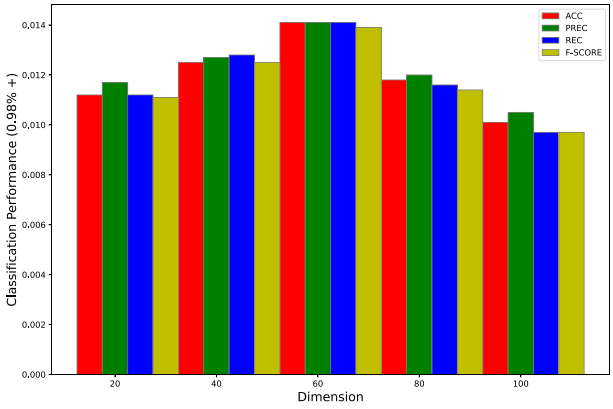


Fig. 4. The effects of dimension expansion for the HALIC dataset using SAE on the accuracy performance of the proposed model.

the same propagation power due to different technological designs, and which are obtained at a very low frequency than expected due to obstacles in the indoor environment.

When an autoencoder step is used, it is seen that the accuracy values that are obtained from both machine and deep learning methods for HALIC notably increases. However, no significant effect was observed for the RFKON and UJIIndoorLoc datasets. The most important reason for this is that SAE expands the limited-length feature space, providing a wider scope for the application of deep learning models' operations. Moreover, the reference points in the HALIC dataset have more similar fingerprints compared to other datasets. The Building 0 in UJIIndoorLoc dataset contains 256 reference points collected from an area of 1600 m^2 [29]. The RFKON dataset contains 54 reference points collected from an area of 800 m^2. The grids are quite far for RFKON and UJIIndoorLoc. However, for HALIC dataset, there exist 123 reference points determined for an area of 492 m^2. Thus, fingerprints are quite close to each other for HALIC dataset. According to the results obtained, we can say that while SAE expands the feature space, it also produces more distinctive representations for reference points that are very close to each other. The results of the dimension expansion using SAE for the HALIC dataset are given Fig. 4 as the change in the accuracy values of the proposed model according to the selected size increase amount. The best accuracy results are obtained for our method when the dimension is equal to 60.

According to Tables 6–8, the highest classification results are obtained by the proposed method. The results show that the proposed hybrid deep model outperforms the baseline methods in terms of accuracy, precision, recall, and F-score. The possible reason for the success of the proposed method is that it leverages two different main layers, CNN to extract local patterns and LSTM to capture long-term correlations. By creating a combined enriched feature space containing information from two different angles, it enables to reveal the more discriminative patterns in indoor position prediction.

Our method achieves higher accuracy results from the other methods for floor classification. Compared to the reference points classification, considerably floor classification provides higher results for UJIIndoorLoc. The most important reason for this is that the changes in the signals received from the APs occur more clearly on the floors. Fig. 5 shows the variation of RSSI signals received from APs for a selected reference point. For the area labeled with SpaceID number 102 in the 0th building in the UJIIndoorLoc dataset, the signal values read from each floor of this area are presented. We selected only twenty observations for this illustration. Although the relevant area has the same area feature (for example, office, kitchen, etc.), it is located on different floors. Signals were received from a total of 67 access points on 4 floors in the relevant area. Although no signal was received from access point 13 on the 0., 1. and 2. floors, RSSI values above –80 dB were read on the 3rd floor. Considering access point number 452, although RSSI values are read from this access point on each floor, it is seen that its value is equal to 0 for some observations. This may be due to environmental factors. In particular, considering that RSSI values over –90 dB are read from this access point on the 3rd floor, it may result that this access point is within this floor. However, since signal values are not obtained in many observations, it can be concluded that an obstacle weakens the signal considerably.

To show the loss and accuracy performances of the proposed architecture over 200 epochs, Figs. 6 and 7 are presented. The figures contain the results of both the training and validation datasets. For all datasets, the loss values of the models decrease, and the accuracy values of the models increase when the number of epochs increases. However, it is seen that there is a convergence in test loss and accuracy values after approximately the 150th epoch. In general, the validation loss

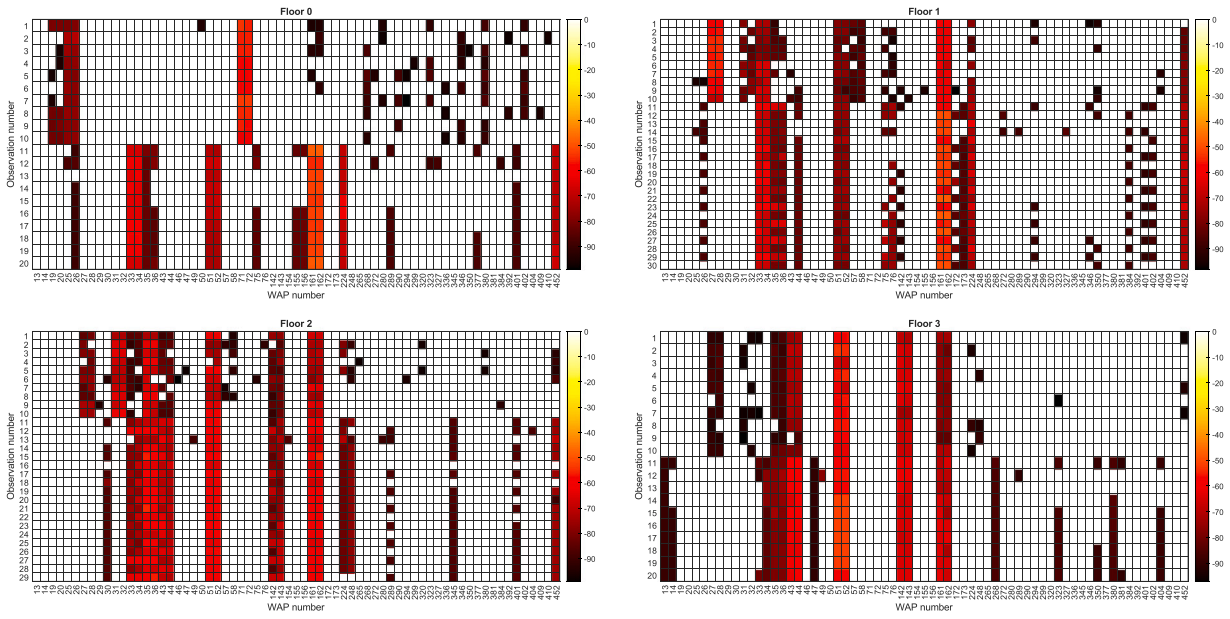
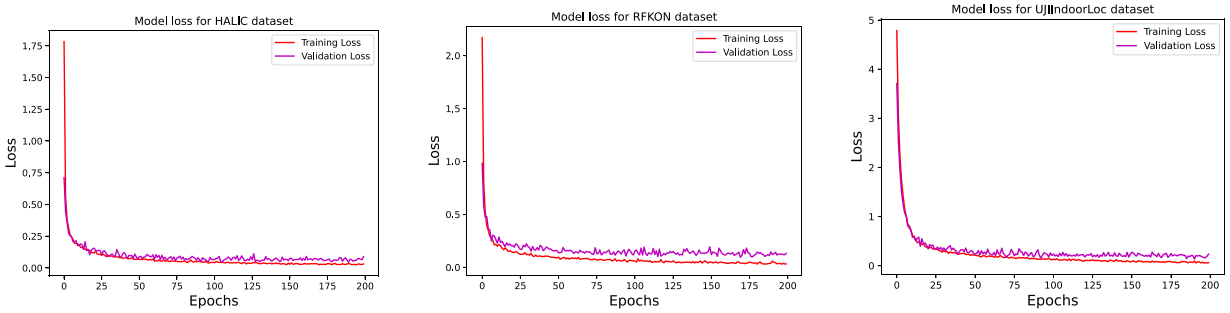
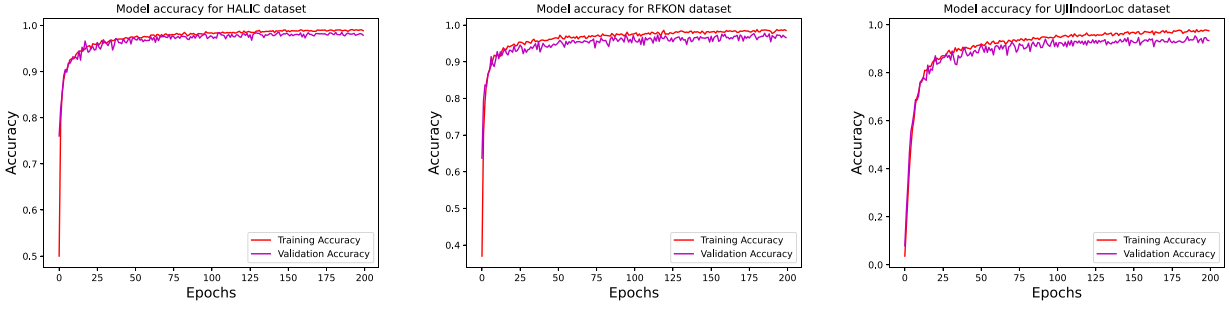


Fig. 5. Variation of RSSI signals received from APs according to floors for a selected reference point.



(a) Training/Validation Loss for HALIC (b) Training/Validation Loss for RFKON (c) Training/Validation Loss for UJIIndoorLoc

Fig. 6. The loss performances of the proposed model.



(a) Training/Validation Loss for HALIC (b) Training/Validation Loss for RFKON (c) Training/Validation Loss for UJIIndoorLoc

Fig. 7. The accuracy performances of the proposed model.

value was observed to be higher than the training loss value. Figs. 6 and 7 show that there does not exist an over-fitting condition within 200 epochs for all three datasets.

To measure the reliability of the model even when very few samples are used for training, Kappa scores of the proposed method for different datasets are given in Fig. 8 [49]. To train the classifier, the training rate (TR) was set as percentiles of 20, 40, 60, and 80, respectively, and the remaining samples were used as the test set. The results show that the proposed method achieves high kappa values for the entire dataset even when very few samples are used for training (20 percent).

4.3.5. Comparison with other existing methods

We compare the proposed method with some state-of-the-art methods in the literature. The comparison results of the studies based on the classification performances are presented in Table 10. The proposed method provides the highest localization performances for

Table 10
Accuracy performance results of the proposed method and other methods in the literature.

Study	Method	Dataset	Classification	Result
Turgut et al. [50]	Stacked autoencoder	RFKON	Grid based	0.9595
Keser et al. [51]	Optimal decision tree	RFKON	Grid based	0.8773
Keser et al. [52]	Extreme learning machine	RFKON	Grid based	0.72
Proposed method	Explainable Hybrid deep learning	RFKON	Grid based	0.9842
Turgut et al. [9]	Stacked sparse autoencoder	HALIC	Grid based	0.7260
Proposed method	Explainable hybrid deep learning	HALIC	Grid based	0.9852
Bozkurt et al. [53]	Sequential minimal optimization	UJIIndoorLoc	Grid based	0.81
Akram et al. [54]	Gaussian mixture model soft clustering + Random decision forest	UJIIndoorLoc	Grid based	0.83
Proposed method	Explainable hybrid deep learning	UJIIndoorloc	Grid based	0.9533
Kim et al. [55]	Stacked autoencoder + DNN	UJIIndoorLoc	Floor based	0.9298
Elmokhtar et al. [56]	Recurrent neural networks	UJIIndoorloc	Floor based	0.9523
Qin et al. [57]	Convolutional denoising autoencoder + CNN	UJIIndoorLoc	Floor based	0.953
Seçkin and Çoşkun [58]	Feature generation + Hierarchical fusing machine learning	UJIIndoorLoc	Floor based	0.96
Zhang and Xu [11]	Stacked denoising autoencoders + MLP	UJIIndoorLoc	Floor based	0.9874
Alitaleshi et al. [59]	Extreme learning Machine autoencoder + CNN	UJIIndoorLoc	Floor based	0.9894
Singh et al. [60]	PCA + XGBoost	UJIIndoorLoc	Floor based	0.992
Etiabi et al. [61]	Federated learning + Hierarchical MLP	UJIIndoorLoc	Floor based	0.9955
Proposed method	Explainable hybrid deep learning	UJIIndoorloc	Floor based	0.9995

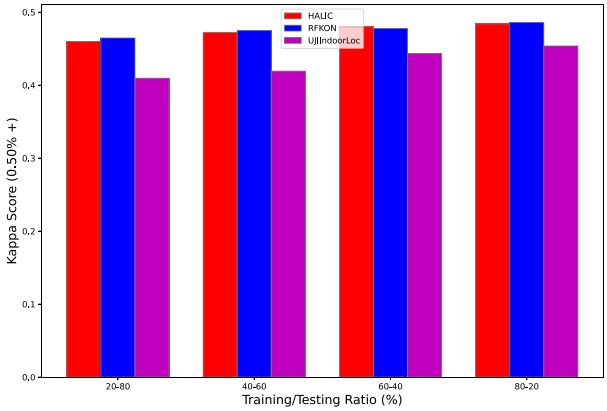


Fig. 8. Kappa values of the proposed model for datasets according to different training and testing rates.

all datasets when compared to competitive indoor localization methods. For RFKON dataset, the second highest result is obtained by the method [50] that implements a stacked autoencoder for feature extraction phase. The authors [50] emphasize that they achieved higher localization performances by expanding the feature space that consists of a few APs using stacked autoencoder. However, in the learning phase after the feature extraction process, they use a simple multi layer perceptron. The other classification performances were presented by the authors in the papers [51,52]. They use optimal decision tree and extreme machine learning for indoor localization. For HALIC dataset, the proposed method significantly achieves better localization performances than the method that uses a sparse autoencoder [9,62] and MLP. Our method improves the performance by at least 36%. While extreme learning machines provide faster learning, generally deep neural networks can achieve higher accuracy for highly nonlinear data. The possible reason why the proposed model obtains higher results from these methods may be that it applies a learning process over many hidden layers using both CNN and LSTM, compared to decision trees, extreme learning machines, and MLP. Moreover, it is observed that the filtering process used in the proposed method is an important factor in increasing the accuracy performances.

UJIIndoorLoc is one of the fundamental datasets used extensively in indoor positioning studies. Studies in the literature provide classification-based approaches and perform position determination based on x and y coordinates using the relevant dataset. It is possible to realize building, floor, and space-based positioning on the UJIIndoorLoc dataset. This study presents building, floor, and grid (space) based localization performances. Accordingly, with the proposed method, building detection is determined with 100% accuracy. Recent studies based on grid-based (room or space) and floor-based classification in UJIIndoorLoc are included in the Table 10. There are very few studies in the literature on grid-based classification. Our method outperforms the methods that use Sequential Minimal Optimization (SMO), and Random Decision Forest based on Gaussian Mixture Model (GMM) Soft Clustering. SMO may not produce very effective results for highly nonlinear data. For the UJIIndoorLoc dataset with a large number of AP points, and RP points, a lower accuracy rate may therefore be achieved. With GMM clustering, the authors aim to reveal subsets containing similar observations. However, since the Wi-Fi propagation characteristics and GMM distributions are not always perfectly aligned, the key attributes of the data may not be captured accurately, although the approach aims to distinguish different RP groups.

Recent studies based on floor estimation for UJIIndoorLoc are included in the Table 10. There are three different buildings in the related dataset and a total of 13 floors, with 4 or fewer floors in each building. In the literature, there are studies on the determination of floors in a single building, as well as studies on the determination of a total of 13 floors with a multi-label approach. In order to make an inclusive comparison, a study for multi-label classification and the detection of 13 floors is carried out in this study. The results demonstrate that our method achieves better accuracy results than the state-of-the-art methods. The second highest result is obtained by the method [61] that uses hierarchical deep learning architecture. The model aim to capture the hierarchy between floors and buildings. The third highest result is obtained by the method [60] that uses PCA for handling sparsity, reducing dimensionality, and removing noise, and XGBoost for the learning phase. The most important possible reason why our method produces more successful results than these methods is the particle filter used in the preprocessing process and making an effective mapping. In addition, it uses the advantage of two different architectures CNN and LSTM on the features obtained as a result of filtering, and performs both local and global learning. In this respect, it is more capable of distinctive feature engineering than XGBoost and MLP architecture. Table 10 also presents the methods for obtaining localization estimation by reducing noise and size, with different preprocessing processes as stacked autoencoder, denoising autoencoder and Extreme Learning Machine autoencoder as preprocessing steps. However, it can be difficult for the denoising autoencoder to choose an appropriate noise level if the data was obtained quite noisy. At this stage, the Stacked Autoencoders may produce better results as they do not involve adding noise during training. However, if the observed signal values in WiFi signals are quite small due to noise, the stacked autoencoder may reduce the importance of low-frequency features in the layer hierarchy.

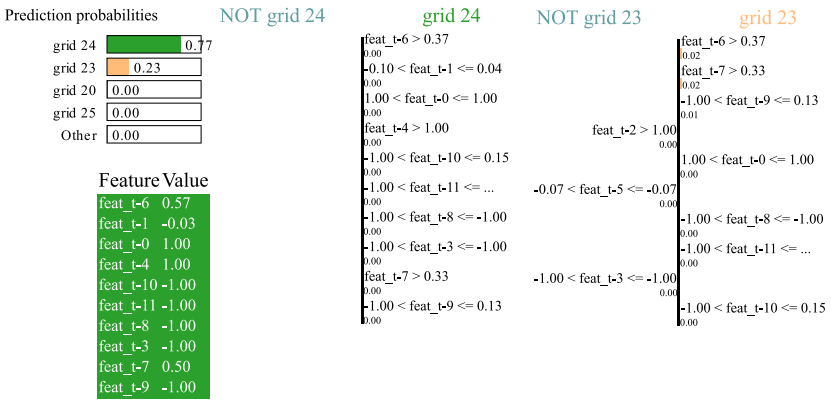


Fig. 9. The local explanations generated from LIME for a given test sample in HALIC dataset.

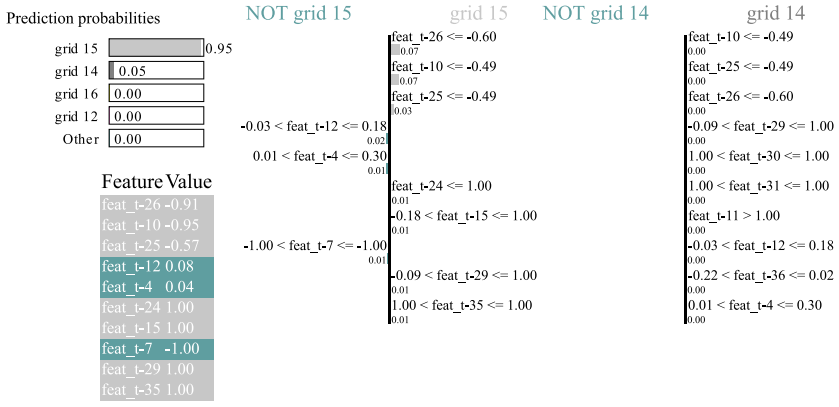


Fig. 10. The local explanations generated from LIME for a given test sample in RFKON dataset.

At this stage, the proposed method rearranges the noisy observed signal values as different from the expected WiFi values for the reference point according to the values obtained in the previous and next observation using the particle filter.

4.4. Explainable indoor localization results

SHAP and LIME techniques were used in the proposed approach and the contribution of the location of the access points to the regional determination was measured. In the tests carried out, adjacent areas, areas not adjacent to each other but on the same floor, and areas on different floors were examined.

The results obtained with the proposed model were analyzed using LIME for three datasets. The local explanations of the model for the datasets are presented in Figs. 9–11, respectively. Only one sample from all three datasets was chosen for generating LIME explanations. Accordingly, the results obtained by LIME include the features and their values that enable the sample selected as a test to be classified into existing grids (reference points).

Considering the result of the HALIC dataset given in Fig. 9, it is seen that the related sample is assigned to grid 24 with the highest probability. Since the ground-truth label of the selected sample is grid 24, the model obtains a correct classification result. The second highest probability for that sample is obtained for grid 23. Since grid 23 and grid 24 are reference points located very close to each other, some similarities can be expected in the collected AP values. Except for the 2nd, 5th, and 3rd access points, the reference values for the access points listed in grid 23 and grid 24 appear to be in the same range in both grids. This is because the distance between these two reference points is approximately 2 m. In this case, it is concluded that the prediction probabilities obtained by the model are consistent. The top ten features and their values that are decisive in the prediction results for grid 24 and grid 23 are included in Fig. 9. For this example, it is seen that the values of ten different features of this sample are compatible with the expected reference values of grid 24. The selected sample is allocated to grid 24, especially since the values from the 2nd, 5th, and 3rd access points are not compatible with the expected reference values of grid 23.

Similarly, Fig. 10 presents the explanations about feature weights used in the position prediction using a single sample for RFKON. For the given test sample belonging to RFKON dataset, the highest class probability is obtained for grid 15. Although the model finds a very low probability for grid 14, it is observed that the signal attribute values of this sample are compatible with the expected reference attribute values of grid 14. However, here it is concluded that the attribute values, which are decisive for assigning the example to grid 15, are quantitatively closer to the reference attribute values of grid 15. For this example, it can be seen that the access points that differentiate grid 15 from grid 14 are the signal values obtained from 15th, 24th, and 35th access points.

Fig. 11 provides the explanations of classification for a selected sample from UJIIndoorLoc datasets by the proposed method. The selected sample is assigned to grid 0 with the proposed method. It is compatible with the reference values of grid 0 and the reference values of the signal values of the sample. It is seen that the reference points affecting both grid 0 and grid 1 are quite different from each other. In this case, it is concluded that the distance between these two grids is reasonably large.

We also provide the global interpretation of the proposed model using SHAP. The results for grid 1 and grid 70 as adjacent grids in the HALIC dataset are presented in Fig. 12. The related analysis was carried out using 10 samples belonging to these grids. Grid 1 and grid

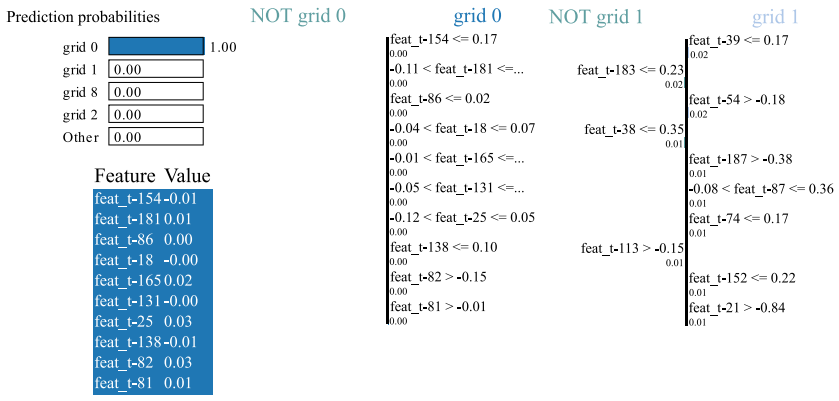
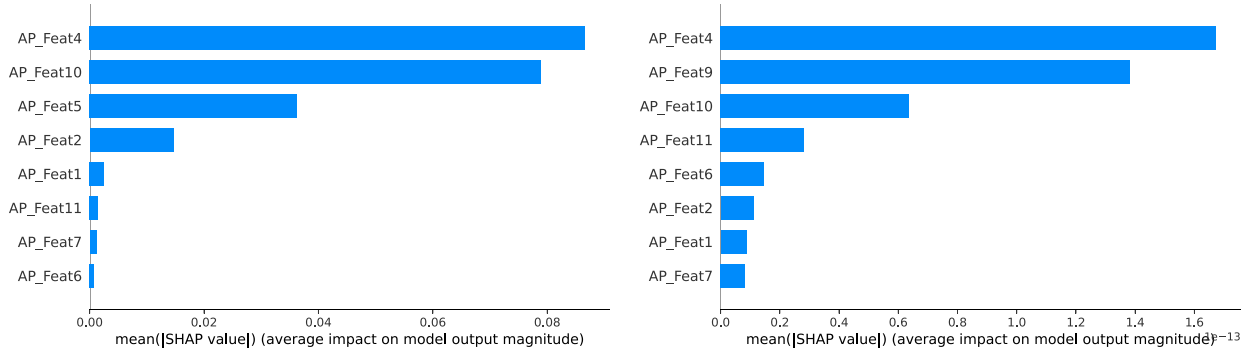
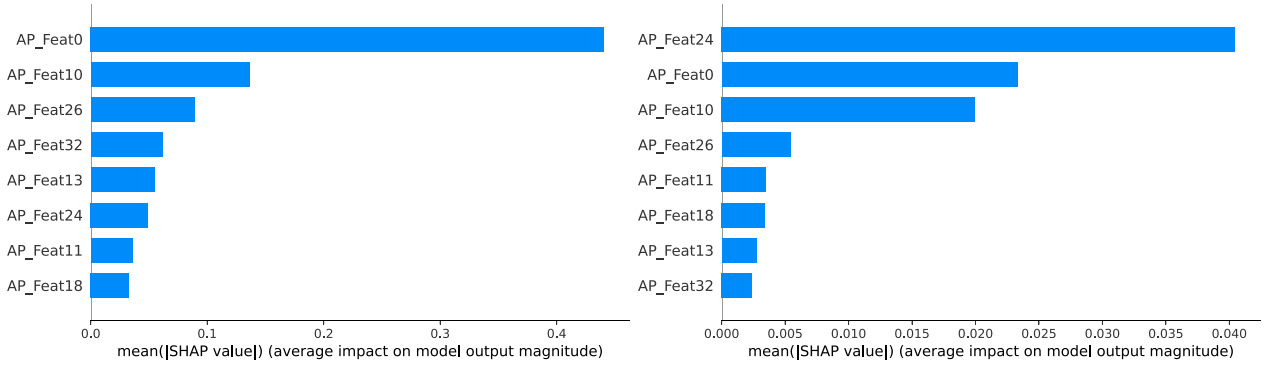


Fig. 11. The local explanations generated from LIME for a given test sample in UJIIndoorLoc dataset.



(a) Grid 1 (b) Grid 70

Fig. 12. The global interpretation of the proposed method for some specific reference points in HALIC dataset using SHAP.

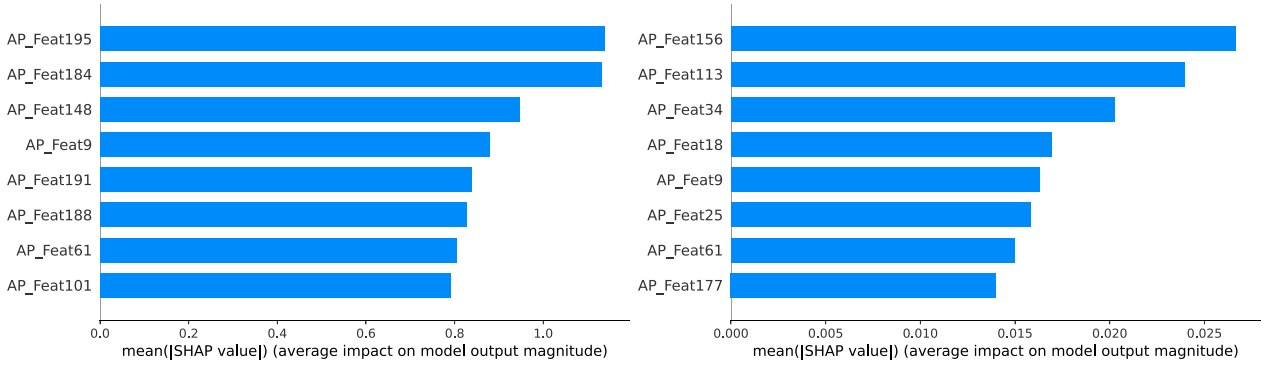


(a) Grid 1 (b) Grid 6

Fig. 13. The global interpretation of the proposed method for some specific reference points in RFKON dataset using SHAP.

70 are not located in the same open office. Grid 1 is in 1nd office, while grid 70 is in 5th office. Access points 0, 1, 2, and 3 are placed in the 1nd office as given in Table 3. In the 5th office, there are three access points numbered as 8, 9, 10. In the classification of the ten selected samples, it is seen that the access points that are more distinctive for grid 1 are 4, 10, 5, and 2, respectively. It is observed that only AP 2 is effective from the access points in the room where grid 1 is located. On the other hand, APs in the 3rd (AP 4), 4th (AP 5), and 5th office (AP 10) seem to be more effective. The most important reason for this may be that similar signal values are obtained from the APs in this office for almost all the grids in the 1st office. In this case, the signal values that are distinctive for the grids in this office are taken from other APs in the vicinity. For grid 70, similarly, the most distinctive access point is AP 4, located in the 3rd office. However, the 9th and 10th APs in the same room as this grid are also effective in classifying the samples.

In Fig. 13, the proposed model is explained on the RFKON dataset by using the SHAP model in grids located in the same building but not adjacent to each other. Accordingly, the results for Grid 1 and grid 6 in the RFKON dataset are presented. Grid 1 and grid 6 are located in the same floor. As expected, there are intersections between the access points that offer the most important features in areas within the same building. A total of 7 APs are common, which are decisive for grid 1 and grid 6. However, the effectiveness of these APs in separating grids differs. For instance, while AP 24 is the most effective access point for grid 6, the impact of this access point for grid 1 is lower than other APs.



(a) Grid 1 (b) Grid 5

Fig. 14. The global interpretation of the proposed method for some specific reference points in UJIIndoorloc dataset using SHAP.

Table 11
Comparisons of model size and complexity.

Methods	Without SAE				With SAE			
	Flops	Params	Model size	Testing time	Flops	Params	Model size	Testing time
DNN	238×10^4	1190k	4663 kB	15×10^{-2} ms	114×10^4	719k	2851 kB	14×10^{-2} ms
CNN	184×10^5	9145k	35,868 kB	35×10^{-2} ms	323×10^4	1608k	6812 kB	18×10^{-2} ms
LSTM	355×10^5	17,701k	69,165 kB	2 ms	526×10^4	2628k	11,181 kB	68×10^{-2} ms
CNN+LSTM	185×10^5	9186k	35,904 kB	1.3 ms	329×10^4	1649k	6848 kB	64×10^{-2} ms
Our method					227×10^4	1126k	6734 kB	69×10^{-2} ms

In addition, the results using SHAP on UJIIndoorLoc are presented in Fig. 14. The ten APs that are most decisive in allocating selected samples to grid 1 or grid 5 are given. These grids are located on different floors. The results show that there are only two access points (AP 9 and AP 61) common to the most efficient APs listed for these grids. Except for these APs, the APs that are effective for separating these two grids are quite different. The most important reason for this is that the signal values received from the APs differ significantly at different floors since these grids are located on different floors.

4.5. Analysis of computational complexity

To show the efficiency of the models, the floating point operations per second (Flop), number of parameters (Params), and model size of the models are given in Table 11. These complexity results are reported for the UJIIndoorLoc dataset since this dataset has a larger input and output size when compared to the other datasets. All parameter settings are used the same, only the batch size is set to 1 for flops calculation. The overall complexity of the models may vary depending on the input size, number of layers, layer types (CNN, LSTM or pooling), number of dense layers, number of neurons, and output layer. The results given in Table 11 show that the baseline methods without SAE have higher FLOPs, the number parameters, and model size when compared to the results obtained from the models with SAE. This is because the input size is equal to the number of sensors. Since the UJIIndoorLoc dataset has 520 sensor data, it increases the number of floating-point calculations of models and increases the connections between layers. As we reduce the input size to 60 using SAE, naturally the models have lower Flops, number of parameters, and model size. Table 11 shows that DNN has the smallest FLOPs, the number of parameters, and model size. This is because DNN contains only dense layers and softmax layers. The performance of DNN is followed by that of our method. Since our method has two convolutional and pooling layers, the size of the feature maps generated within the model is reduced in several steps. In this case, the input size transmitted to the dense layer is lower than the CNN, LSTM, and CNN-LSTM models.

The testing times of the models for a single sample are given in Table 11. Since the filtering process is applied for each new incoming data, the time required for the filtering process is also important. The filtering process of one sample is approximately 0.010 s Considering the filtering and testing times, the proposed architecture has a high potential to be used for real-time indoor localization.

5. Conclusion

In this study, an explainable hybrid deep learning architecture is presented to provide mobility management in indoor areas, one of the fundamental services in the Internet of Things ecosystem. In order to test the proposed method, three different datasets containing RSSI data collected by fingerprint method belonging to WiFi signals were used: HALIC, RFKON, UJIIndoorLoc. In the selected datasets, the signal distortions caused by the effects encountered in indoor areas are removed by using a particle filter, and the original signal values are estimated. Due to the different forms of the network structures formed in the studies carried out using the signal map, all network types; in order to create an approach that can cover overcomplete and undercomplete networks, the features of the datasets are evaluated using sparse autoencoders. The proposed architecture uses a sparse autoencoder for feature extraction and then applies Convolutional Neural Network (CNN) and Long-Short-Term Memory (LSTM) simultaneously for indoor localization. We present the effectiveness of the proposed hybrid model by providing comprehensive benchmarks using different deep learning architectures such as DNN, CNN, LSTM. The highest positioning accuracy was achieved using our approach with 98.52 in the HALIC dataset, 98.42 in the RFKON dataset, and 95.33 in the UJIIndoorLoc dataset. In order to analyze the effects of the positions of the APs and values of the APs on the position estimation of the proposed model, the results from two XAI methods as LIME and SHAP are presented.

We focus our future work on two aspects. Firstly, we plan to transform our method for the problem of estimating three-dimensional location coordinates. Secondly, we aim to improve the proposed deep learning model by applying different filtering processes in real time.

CRediT authorship contribution statement

Zeynep Turgut: Conceptualization, Methodology, Investigation, Resources, Software, Writing – original draft. **Arzu Gorgulu Kakisim:** Conceptualization, Methodology, Investigation, Resources, Software, Writing – original draft.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] K. Ashton, That 'internet of things' thing, in: That 'Internet of Things' Thing-RFID Journal, Vol. 22, 2009, pp. 97–114, URL http://www.rfidjournal.com/article/print/4986.

[2] F. Zafari, A. Gkelias, K.K. Leung, A survey of indoor localization systems and technologies, IEEE Commun. Surv. Tutor. 21 (3) (2019) 2568–2599.

[3] L. Yang, H. Chen, Q. Cui, X. Fu, Y. Zhang, Probabilistic-KNN: A novel algorithm for passive indoor-localization scenario, in: 2015 IEEE 81st Vehicular Technology Conference (VTC Spring), IEEE, 2015, pp. 1–5.

[4] Z. Turgut, G.Z.G. Aydin, A. Sertbas, Indoor Localization Techniques for Smart Building Environment, Vol. 83, Elsevier, 2016, pp. 1176–1181, http://dx.doi.org/10.1016/j.procs.2016.04.242.

[5] M. Maduranga, R. Abeysekera, TreeLoc: an ensemble learning-based approach for range based indoor localization, Int. J. Wirel. Microw. Technol. (IJWMT) 11 (5) (2021) 18–25.

[6] F. Zafari, S. Member, A. Gkelias, S. Member, K.K. Leung, A survey of indoor localization systems and technologies, IEEE Commun. Surv. Tutor. 21 (3) (2019) 2568–2599.

[7] S. Ustebay, Z. Turgut, O.C. Turna, M. Ali Aydin, T.B. Atmaca, Analysis of device-free and device dependent signal filtering approaches for indoor localization based on Earth's magnetic field system, in: CEUR Workshop Proceedings, Vol. 2533, 2019, pp. 1–13, URL www.scopus.com.

[8] J. Cha, E. Lim, A hierarchical auxiliary deep neural network architecture for large-scale indoor localization based on Wi-Fi fingerprinting, Appl. Soft Comput. 120 (2022) 108624, http://dx.doi.org/10.1016/j.asoc.2022.108624.

[9] Z. Turgut, S. Üstebay, M. Ali Aydın, G.Z. Gürkaş Aydın, A. Sertbaş, Performance analysis of machine learning and deep learning classification methods for indoor localization in Internet of things environment, Trans. Emerg. Telecommun. Technol. 30 (9) (2019) e3705.

[10] Z.E. Khatab, A. Hajihoseini, S.A. Ghorashi, A fingerprint method for indoor localization using autoencoder based deep extreme learning machine, IEEE Sensors Lett. 2 (2017) 1–4, http://dx.doi.org/10.1109/lsens.2017.2787651.

[11] H. Zhang, X. Jiachuan, Indoor Wi-Fi fingerprint localization based on SDAE and MLP with self-attention mechanism, in: Chinese Automation Congress (CAC), IEEE, 2022, pp. 1963–1967.

[12] E. Ebaid, N. Keivan, Optimum NN algorithms parameters on the ujiIndoorLoc for Wi-Fi fingerprinting indoor positioning systems, in: 32nd International Telecommunication Networks and Applications Conference (ITNAC) 2022, IEEE, 2022, pp. 280–286.

[13] J. Xue, J. Liu, M. Sheng, Y. Shi, J. Li, A WiFi fingerprint based high-adaptability indoor localization via machine learning, China Commun. 17 (7) (2020) 247–259.

[14] B. Jia, W. Qiao, Z. Zong, S. Liu, M. Hijji, J.D. Ser, K. Muhammad, A fingerprint-based localization algorithm based on LSTM and data expansion method for sparse samples, Future Gener. Comput. Syst. 137 (2022) 380–393, http://dx.doi.org/10.1016/j.future.2022.07.021.

[15] X. Liu, B. Zhou, P. Huang, W. Xue, Q. Li, J. Zhu, L. Qiu, Kalman filter-based data fusion of wi-fi rtt and pdr for indoor localization, IEEE Sens. J. 21 (6) (2021) 8479–8490.

[16] G. Raja, S. Suresh, S. Anbalagan, A. Ganapathisubramaniyan, N. Kumar, PFIN: An efficient particle filter-based indoor navigation framework for UAVs, IEEE Trans. Veh. Technol. 70 (5) (2021) 4984–4992.

[17] X. Song, X. Fan, C. Xiang, Q. Ye, L. Liu, Z. Wang, X. He, N. Yang, G. Fang, A novel convolutional neural network based indoor localization framework with WiFi fingerprinting, IEEE Access 7 (2019) 110698–110709, http://dx.doi.org/10.1109/ACCESS.2019.2933921.

[18] N. Hernández, I. Parra, H. Corrales, R. Izquierdo, A.L. Ballardini, C. Salinas, I. García, WiFiNet: WiFi-based indoor localisation using CNNs, Expert Syst. Appl. 177 (2021) http://dx.doi.org/10.1016/j.eswa.2021.114906.

[19] K.S. Kim, S. Lee, K. Huang, A scalable deep neural network architecture for multi-building and multi-floor indoor localization based on Wi-Fi fingerprinting, Big Data Anal. 3 (2018) http://dx.doi.org/10.1186/s41044-018-0031-2.

[20] Z. Chen, H. Zou, J.F. Yang, H. Jiang, L. Xie, WiFi fingerprinting indoor localization using local feature-based deep LSTM, IEEE Syst. J. 14 (2020) 3001–3010, http://dx.doi.org/10.1109/JSYST.2019.2918678.

[21] F. Orujov, W. Wei, Y. Li, Smartphone based intelligent indoor positioning using fuzzy logic, Future Gener. Comput. Syst. 89 (2018) (2018) 335–348, http://dx.doi.org/10.1016/j.future.2018.06.030.

[22] L. Li, X. Guo, Y. Zhang, N. Ansari, H. Li, Long short-term indoor positioning system via evolving knowledge transfer, IEEE Trans. Wirel. Commun. 21 (7) (2022) 5556–5572.

[23] B. Sulaiman, E. Natsheh, S. Tarapiah, Towards a better indoor positioning system : A location estimation process using artificial neural networks based on a semi-interpolated database, Pervasive Mob. Comput. 81 (2022) 101548, http://dx.doi.org/10.1016/j.pmcj.2022.101548.

[24] G. Chen, X. Guo, K. Liu, X. Li, J. Yang, RWKNN : A modified WKNN algorithm specific for the indoor localization problem, IEEE Sensors J. 22 (7) (2022) 7258–7266.

[25] E.S. Lohan, J. Torres-Sospedra, H. Leppäkoski, P. Richter, Z. Peng, J. Huerta, Wi-Fi crowdsourced fingerprinting dataset for indoor positioning, Data 2 (4) (2017) 32.

[26] C. Kumar, S. Member, K. Rajawat, Dictionary-based statistical fingerprinting for indoor localization, IEEE Trans. Veh. Technol. 68 (9) (2019) 8827–8841, http://dx.doi.org/10.1109/TVT.2019.2929360.

[27] X. Guo, N. Ansari, Indoor localization by fusing a group of fingerprints based on random forests, IEEE Int. Things J. 5 (6) (2018) 4686–4698.

[28] H. Li, Z. Qian, C. Tian, X. Wang, TILoc : Improving the robustness and accuracy for fingerprint-based indoor localization, IEEE Int. Things J. 7 (4) (2020) 3053–3066.

[29] J. Torres-Sospedra, R. Montoliu, A. Martínez-Usó, J.P. Avariento, T.J. Arnau, M. Benedito-Bordonau, J. Huerta, UJIIndoorLoc: A new multi-building and multi-floor database for WLAN fingerprint-based indoor localization problems, in: 2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN), IEEE, 2014, pp. 261–270.

[30] Y. Huo, P. Puspitaningayu, N. Funabiki, K. Hamazaki, M. Kuribayashi, K. Kojima, A proposal of the fingerprint optimization method for the fingerprint-based indoor localization system with IEEE 802.15.4 devices, Information 12 (5) (2022) 211.

[31] J. Xue, J. Liu, M. Sheng, Y. Shi, J. Li, A WiFi fingerprint based high-adaptability indoor localization via machine learning, China Commun. 17 (J7) (2020) 247–259.

[32] G.M. Mendoza-Silva, A.C. Costa, J. Torres-sospedra, M. Painho, J. Huerta, Environment-aware regression for indoor localization based on wifi fingerprinting 22 (6) (2022) 4978–4988.

[33] G.M. Mendoza-silva, P. Richter, Long-term WiFi fingerprinting dataset for research on robust indoor positioning, Data 3 (1) (2018) 1–17, http://dx.doi.org/10.3390/data3010003.

[34] T. King, S. Kopf, T. Haenselmann, C. Lubberger, W. Effelsberg, CRAWDAD dataset mannheim/compass (V. 2008-04-11), 2008, URL https://crawdad.org/mannheim/compass/20080411,[Online]..

[35] H.-Y. Hsieh, S.W. Prakosa, J.-S. Leu, Towards the implementation of recurrent neural network schemes for WiFi fingerprint-based indoor positioning, in: 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall), IEEE, 2018, pp. 1–5.

[36] H. Obeidat, W. Shuaieb, O. Obeidat, R. Abd-Alhameed, A review of indoor localization techniques and wireless technologies, Wirel. Pers. Commun. 119 (1) (2021) 289–327.

[37] W. Li, J. Wang, Magnetic sensors for navigation applications: an overview, J. Navig. 67 (2) (2014) 263–275.

[38] D. Charte, F. Charte, S. García, M.J. del Jesus, F. Herrera, A practical tutorial on autoencoders for nonlinear feature fusion: Taxonomy, models, software and guidelines, Inf. Fusion 44 (2018) 78–96, http://dx.doi.org/10.1016/j.inffus.2017.12.007.

[39] F. Gustafsson, Particle filter theory and practice with positioning applications, IEEE Aerosp. Electr. Syst. Mag. 25 (7) (2010) 53–82.

[40] A. Ng, et al., Sparse Autoencoder, Vol. 72, in: CS294A Lecture notes, 2011, pp. 1–19, (2011).

[41] F. Strub, J. Mary, P. Philippe, Collaborative filtering with stacked denoising autoencoders and sparse inputs, in: NIPS Workshop on Machine Learning for ECommerce, 2015.

[42] S. Malek, F. Melgani, Y. Bazi, One-dimensional convolutional neural networks for spectroscopic signal regression, J. Chemometr. 32 (5) (2018) e2977.

[43] W. Samek, G. Montavon, A. Vedaldi, L.K. Hansen, K.-R. Müller, Explainable AI: Interpreting, Explaining and Visualizing Deep Learning, Vol. 11700, Springer Nature, 2019.

[44] M.T. Ribeiro, S. Singh, C. Guestrin, Model-agnostic interpretability of machine learning, 2016, arXiv preprint arXiv:1606.05386.

[45] Z. Abou El Houda, B. Brik, L. Khoukhi, "Why should I trust your IDS?": An explainable deep learning framework for intrusion detection systems in internet of things networks, IEEE Open J. Commun. Soc. 3 (2022) 1164–1176.

[46] S.M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: Advances in neural information processing systems, Vol. 30, 2017.

[47] S.B. Keser, U. Yayan, A. Yazici, S. Gunal, International Journal of Computer Science: Theory and Application A priori verification and validation study of RFKON database, Int. J. Comput. Sci. Theor. App 5 (2016) 20–27, URL www.orb-academic.org.

[48] G.M. Mendoza-Silva, P. Richter, J. Torres-Sospedra, E.S. Lohan, J. Huerta, Long-term WiFi fingerprinting dataset for research on robust indoor positioning, Data 3 (1) (2018) 3.
[49] J. Sim, C.C. Wright, The kappa statistic in reliability studies : Use , interpretation , and, Phys. Ther. 85 (3) (2005) 257–268.
[50] Deep Learning in Indoor Localization Using Wifi, in: Lecture Notes in Electrical Engineering, 504, 2019, http://dx.doi.org/10.1007/978-981-13-0408-8_9.
[51] S.B. Keser, U. Yayan, A case study of optimal decision tree construction for RFKON database, in: Nternational Symposium on INnovations in Intelligent SysTems and Applications, IEEE, 2016, pp. 1–6.
[52] S.B. Keser, A. Yazici, S. Gunal, A hybrid fingerprint based indoor positioning with extreme learning machine, in: 25th Signal Processing and Communications Applications Conference (SIU), 2017, pp. 14–17.
[53] S. Bozkurt, G. Elibol, S. Gunal, U. Yayan, A comparative study on machine learning algorithms for indoor positioning, in: 2015 International Symposium on Innovations in Intelligent SysTems and Applications (INISTA), IEEE, 2015, pp. 1–8.
[54] B.A. Akram, A.H. Akbar, O. Shafiq, HybLoc: Hybrid indoor Wi-Fi localization using soft clustering-based random decision forest ensembles, IEEE Access 6 (2018) 38251–38272.
[55] K.S. Kim, S. Lee, K. Huang, A scalable deep neural network architecture for multi-building and multi-floor indoor localization based on Wi-Fi fingerprinting, Big Data Anal. 3 (2018) 1–17.
[56] A. Elmokhtar, A. Elesawi, K.S. Kim, Hierarchical multi-building and multi-floor indoor localization based on recurrent neural networks, in: 2021 Ninth International Symposium on Computing and Networking Workshops (CANDARW) 2021, IEEE, 2021, pp. 2021–2024, http://dx.doi.org/10.1109/CANDARW53999.2021.00038.
[57] F. Qin, T. Zuo, X. Wang, Ccpos: Wifi fingerprint indoor positioning system based on cdae-cnn, Sensors 21 (4) (2021) 1114.
[58] A.Ç. Seçkin, A. Coşkun, Hierarchical fusion of machine learning algorithms in indoor positioning and localization, Appl. Sci. 9 (18) (2019) 3665.
[59] A. Alitaleshi, H. Jazayeriy, J. Kazemitabar, EA-CNN: A smart indoor 3D positioning scheme based on Wi-Fi fingerprinting and deep learning, Eng. Appl. Artif. Intell. 117 (2023) 105509.
[60] N. Singh, S. Choe, R. Punmiya, N. Kaur, XGBLoc: XGBoost-based indoor localization in multi-building multi-floor environments, Sensors 22 (17) (2022) 6629.
[61] Y. Etiabi, W. Njima, E.M. Amhoud, Federated learning based hierarchical 3D indoor localization, in: 2023 IEEE Wireless Communications and Networking Conference (WCNC), IEEE, 2023, pp. 1–6.
[62] Z. Turgut, Nesnelerin İnterneti için Hareketlilik Yönetimi (Ph.D. thesis), İstanbul Üniversitesi, 2018.

Dr. Zeynep Turgut is currently an Assistant Professor at the Department of Computer Engineering, Istanbul Medeniyet University, Turkey. She received her Ph.D. degree from the Department of Computer Engineering, Istanbul University, Turkey, in 2018. Her primary research interests are computer networks, indoor localization, and intrusion detection.

Dr. Arzu Gorgulu Kakisim is currently an Assistant Professor at the Department of Computer Engineering, Istanbul Medeniyet University, Turkey. She received her Ph.D. degree from the Department of Computer Engineering, Gebze Technical University, Turkey, in 2019. Her primary research interests are representation learning and community search for heterogeneous networks, deep learning and malware detection.