Engineered Science
DOI: https://dx.doi.org/10.30919/es1289

# Machine Learning-Based WiFi Indoor Localization with FasterKAN: Optimizing Communication and Signal Accuracy

Yihang Feng,[1] Yi Wang,[1] Bo Zhao,[2] Jinbo Bi[2,*] and Yangchao Luo[1,*]

## Abstract

WiFi Received Signal Strength Indicator (RSSI) fingerprint has emerged as a powerful indoor localization technique, surpassing other methods such as GPS-based and Bluetooth-based approaches in terms of accuracy and cost-effectiveness. However, challenges persist in WiFi indoor localization, including the need for shorter latency, simpler systems, better scalability, and robustness. In this study, we applied FasterKAN, a variant of Kolmogorov-Arnold Networks (KAN), to address these challenges. FasterKAN significantly shortens model forward time while enhancing localization accuracy. To evaluate its performance, we applied FasterKAN alongside traditional machine learning algorithms to two benchmark datasets: UJIIndoorLoc and SODIndoorLoc. Our results demonstrate remarkable accuracy: 99% for floor and building classification, and 71% for space ID classification. The mean position error (m) is impressively low at 3.56 for the prediction of coordinates. When using graphics processing unit (GPU) accelerating and central processing unit (CPU) only, the model forward time of FasterKAN was about 8.2 and 3.7 times shorter than that of convolutional neural network (CNN) model respectively, with roughly the same number of total parameters. This research contributes to advancing WiFi indoor localization, providing an accurate, robust, and efficient solution for real-world applications such as navigation in large public spaces, optimizing resource management in smart buildings, enhancing operational efficiency in industrial settings.

*Keywords:* Signal; Indoor localization; WiFi; FasterKAN; UJIIndoorLoc; SODIndoorLoc; Received signal strength indicator.
Received: 02 September 2024; Revised: 28 October 2024; Accepted: 29 October 2024.
Article type: Research article.

## 1. Introduction

Indoor localization technology has become crucial in a variety of settings including retail environments, hospitals, and large public spaces like airports and museums. With the rise of the Internet of Things (IoT), the ability to precisely locate devices and individuals indoors plays a pivotal role in enabling smart environment functionalities, enhancing user experience, and improving operational efficiency. Indoor localization has been challenging because the global positioning system (GPS) signal,[1] which serves as a standard solution for outdoor localization, cannot penetrate well in the indoor environment. Traditional GPS technology, while effective outdoors, fails to deliver the same performance indoors due to the lack of direct line-of-sight with satellites,[2] which is crucial for accurate signal reception and processing. This limitation necessitates the development of alternative localization methods that can provide high accuracy in indoor environments.

Indoor localization has benefited from modern technologies, such as visible light, acoustic signal, ultrasound, and radio communication technologies, including Wi-Fi, Bluetooth, ZigBee, radio-frequency identification (RFID), and ultrawideband (UWB).[3-5] Various signal metrics are considered, such as received signal strength indicator (RSSI), channel state information (CSI), Angle of Arrival (AoA), Time Difference of Arrival (TDoA), and Time of Flight (ToF). Among various indoor localization techniques, WiFi fingerprinting with RSSIs from different access points (APs) has been an effective source for indoor localization because the collected WiFi signals vary according to indoor locations, and thus WiFi fingerprinting is a prominent method due to its cost-effectiveness and the ubiquity of WiFi infrastructure.[6] It also has high accuracy, high feasibility, simplicity, and deployment practicability.[1,7] Unlike GPS, WiFi-based methods utilize RSSIs from multiple APs within a building to determine a device's location. This method is not only economical but also leverages existing WiFi networks,

[1] *Department of Nutritional Sciences, University of Connecticut, Storrs, CT 06269, United States.*
[2] *Department of Computer Science and Engineering, University of Connecticut, Storrs, CT 06269, United States.*
**Email:* jinbo.bi@uconn.edu (J. Bi), yangchao.luo@uconn.edu (Y. Luo)

eliminating the need for additional specialized hardware. WiFi fingerprinting usually involves two phases: (1) an offline phase where RSSIs are collected from different APs at many known locations to build a fingerprint database of the environment and (2) an online phase where the position of a target is estimated by comparing the current captured RSSIs with those in the database.

For other indoor localization techniques, Bluetooth localization uses Bluetooth low energy beacons to interact with mobile devices for positioning but requires many beacons, making it costly and complex to set up.[8,9] RFID systems, using signal-emitting tags and readers, are expensive and limited by the need for direct line-of-sight.[10] In contrast, WiFi fingerprinting utilizes existing WiFi infrastructure, significantly cutting down on costs and complexity. Expanding WiFi systems involves simply adding more APs, which are inexpensive and easy to install. WiFi's ubiquity and continuous network improvements enhance localization accuracy and reliability, and WiFi systems generally demand less maintenance than other technologies,[11] being integral to standard IT operations.

In our work, we used two benchmark WiFi RSSI fingerprint datasets to evaluate model performance, *i.e.*, UJIIndoorLoc[12] and SODIndoorLoc.[13] The UJIIndoorLoc dataset is a detailed WiFi localization dataset covering the University of Jaén's indoor spaces with coverage of 108,703 $m^2$, spanning multiple buildings and floors. It includes 19,937 training and 1,111 validation records with 529 attributes such as WiFi fingerprints, coordinates, and other details. The SODIndoorLoc dataset is an open dataset designed as a supplement to the UJIIndoorLoc dataset. It encompasses three buildings and covers 8,000 $m^2$ across various spaces including corridors, office rooms, and meeting rooms. SODIndoorLoc includes 1,630 reference points and 272 testing points, totaling 23,925 samples (21,205 for training and 2,720 for testing).

People spend approximately 80% of their daily lives indoors and 70% of smartphone usage and 80% of data transmission occur indoors.[14] Singh *et al.* wrote an overview of the machine learning based WiFi indoor localization using RSSI fingerprints,[15] which is highly relevant to our study. They thought in the era of the Internet of Things (IoT) and Industry 4.0, conventional techniques, *e.g.* global navigation satellite systems and vision-based systems, are no longer practical to meet the needs of short delay, low design cost, and high performance in indoor localization. They point out that machine learning based indoor localization heavily relies on datasets and evaluate the indoor localization schemes mainly based on location classification accuracy, positioning error, and model robustness, scalability, and complexity. They summarized the current challenges of WiFi-based indoor localization: lack of privacy, lack of standardization (especially lack of benchmark datasets and dedicated standard machine learning algorithms), need for adaptive radio map construction, heterogenicity in devices, high energy consumption, WiFi network not made for localization, and handover delay during WiFi roaming. Various studies emerged to address the above challenges.

A variety of machine learning approaches have been proposed for indoor localization with WiFi RSSIs in IoT. Yang *et al.*[16] proposed a KNN based method by investigating the sensor data from smartphone and user motions to construct the radio map of a floor plan. Tran and Phamadopted the model-based classification approach based on SVMs. However, these methods used hand-crafted features,[17] so they may not utilize the sensing data fully to learn features. A four-layer deep neural network (DNN) was used to extract features from the raw sensing data and estimate locations inby dividing the indoor environment into hundreds of square grids and classifying the target into a grid.[18] Nguyen performed a literature review and compared the performance of the most popular machine learning approaches based on WiFi fingerprinting, e.g., weighted KNN, Naive Bayes, and neural networks.[19] It suggested that if only WiFi RSSIs were used, complex algorithms might not outperform simple ones. Despite the simplicity of the weighted KNN method, it excelled in most fingerprinting techniques, which is why KNN is the most widely used benchmarking algorithm for indoor localization based on Wi-Fi fingerprinting. Deep reinforcement learning (DRL) was also used for single-floor localization in [20]. It proposed a semi-supervised DRL framework as a learning mechanism in support of smart IoT services and experimented in an indoor localization system. An unsupervised navigation and localization method was proposed in [21] to use a DRL algorithm. However, these methods require the floor plan to be represented by small grids beforehand. As discussed before, any change in the location accuracy requirement renders the entire space to be repartitioned to a finer grid, and then models need to be retrained. These methods can be inefficient because the agent must search grid-by-grid to find the target on the floor, which might require hundreds of steps before reaching the target, depending on the initial position. A top-down searching method was designed to provide on-demand resolution in [22] without the need to divide the floor into grids. This method can localize a target with high accuracy, typically within ten steps. Recent advancements in WiFi-based indoor localization have focused on leveraging deep learning and hybrid approaches to enhance accuracy and efficiency. For instance, Lin et al. proposed a robot-aided hybrid deep learning approach that combines supervised, unsupervised, and semi-supervised learning strategies to improve localization and navigation performance in complex indoor environments.[23] Similarly, Ye and Peng introduced an improved Wi-Fi location fingerprinting algorithm that enhances positioning accuracy by integrating an adaptive K-value WKNN algorithm and a grid-based navigation method.[24]

The above studies might not necessarily use UJIIndoorLoc datasets, and to better compare our work with other studies, we discussed the following related works that used datasets including UJIIndoorLoc. Nowicki and Wietrzykowski developed a deep neural network with stacked autoencoders (SAE) to efficiently reduce the feature space to achieve robust and precise classification.[25] They tried to reduce the dimensionality of the input vector from 520 to 256, 128, and 64 in different scenarios. They achieved their best floor and building accuracy of 92% when using an SAE (256-128-64) with classifier (128-128) network architecture. Song et al. also developed a convolutional neural network (CNN) with an SAE (128-64-128) named CNNLoc and evaluated the model performance on multiple datasets including UJIIndoorLoc.[26] CNNLoc achieved 100% building classification and 95% floor classification separately. CNNLoc also had a positioning error of 11.78 m for UJIIndoorLoc. Jang and Hong developed a CNN with a dropout layer to improve the model's sensitivity to the fluctuation of RSSI values and created a 2D WiFi fingerprint based on RSSI values obtained from neighbor Aps.[27] They also used data balancing and ensemble strategy to address the data imbalance in UJIIndoorLoc. They achieved a 95.41% classification accuracy for floor and building for their proposed CNN model. Akram et al. developed a hybrid localization model based on Gaussian Mixture Model (GMM) soft clustering and Random Decision Forest (RDF) ensembles.[28] Their model aimed to predict both the room level and floor & building level localization. The GMM soft clustering found natural data subsets, guided by Akaike Information Criteria and Bayesian Information Criteria, to help cascaded classifiers better learn underlying data dynamics, while RDF improved the generalization of decision trees. They trained their model independently on the three buildings in UJIIndoorLoc and achieved a mean of 85% room-level classification accuracy. It should be noted that limiting the classification of room level location to a specific building during training and testing is a much easier task for the classifier and thus their room level accuracy is not comparable to our work. Their model had a positioning error of 6.29 m over the entire dataset. Qian *et al.* developed two deep learning models, *i.e.* convolutional mixture density recurrent neural network and variational autoencoder-based semi-supervised learning model.[29] The former is designed for indoor next location prediction, while the latter is designed to compute accurate user location in a semi-supervised learning manner since most real-world WiFi fingerprint data are not labeled. Their work wants to solve two problems: (1) To predict the next indoor location of the users. The input is the WiFi RSSI values at the current time points and the target is the coordinates of the users at the next time points. They treat the input data as sequences since UJIIndoorLoc contains a timestamp feature for each sample. (2) To compute the location of the users. The input is the WiFi RSSI values, and the target is the coordinates of the users. For the first problem, they achieved a positioning error of 6.25 ± 0.8 m for the convolutional mixture density gated recurrent unit (CMDGRU) model. For the second problem, which is more comparable to our work, however, the authors scaled the label coordinates, and thus their results could not be directly compared to our work.

Kolmogorov-Arnold Networks (KAN)represent a novel neural network architecture inspired by the Kolmogorov-Arnold representation theorem,[30] which asserts that any continuous multivariable function can be represented as a superposition of single-variable functions. Introduced in April 2024, KAN has quickly gained attention for their flexibility and efficiency in learning complex data patterns. Unlike traditional neural networks, KAN utilizes learnable activation functions on edges rather than fixed activation functions on nodes, enhancing both accuracy and interpretability. Recent studies have demonstrated KAN's potential across various domains. For instance, Hou et al. showcased KAN's superior performance in data fitting and partial differential equation solving [31]. Another study applied KAN to predict liquefaction-induced settlement, outperforming traditional methods like Random Forest.[32] Additionally, the development of KAN 2.0 has further refined the architecture, making it even more robust and scalable for diverse applications.[33] These advancements underscore KAN's promise as a powerful alternative to conventional neural network models.

We are excited to unveil FasterKAN as our main model in this study, which is an advanced variant of the trending KAN. Specifically tackling the unique challenges of WiFi indoor localization including latency, robustness, scalability, and accuracy, FasterKAN stands out by providing significant improvements in model forward time and overall accuracy when compared to conventional Multi-Layer Perceptrons (MLPs) and other traditional machine learning models. Considering FasterKAN's low model complexity and high accuracy, it is perfectly suited for online applications that demand quick and dependable indoor localization. FasterKAN significantly enhances WiFi indoor localization by addressing key limitations of traditional methods. It offers superior scalability, allowing efficient handling of large-scale environments without increased computational demands. The robustness of FasterKAN ensures high accuracy across diverse datasets and dynamic settings, making it reliable for various applications. Additionally, its simplified architecture reduces model complexity and latency, facilitating quick deployment and minimal maintenance. These advancements make FasterKAN an ideal solution for real-world applications. We did not compare the performance between FasterKAN and neural network models that are more complex than a simple six-layer CNN because we focus both on model accuracy and model forward time. Complex neural networks will easily fail to meet the requirements of the millisecond level latency in real-time indoor localization tasks in our daily life.

The major objective of our work is to provide an accurate, robust, and efficient solution for WiFi based indoor localization that can be easily scaled and implemented in various environments without the need for extensive recalibration. By integrating FasterKAN, we aim to address the key limitations of current technologies, such as the high latency and complexity of existing models, while also pushing the boundaries of accuracy in real-world applications. For example, Fig. 1 illustrates the indoor localization based on WiFi access points in an airport. Our contributions are significant in that we not only demonstrate the application of a novel neural network architecture, *i.e*., FasterKAN, but also provide robust evaluation results in terms of positioning error, and floor, building, and space ID classification accuracy, facilitating the performance comparison across similar studies.

## 2 Modelling

### 2.1 Datasets

We used two public mufti-floor and multi-building WiFi RSSI fingerprint datasets in our work, *i.e.*, UJIIndoorLoc and SODIndoorLoc. UJIIndoorLoc is a widely used dataset in WiFi indoor localization studies and SODIndoorLoc is a supplementary dataset of UJIIndoorLoc. An example of samples and features in UJIIndoorLoc is in Table 1. There are three independent sub-datasets of SODIndoorLoc named by the building CETC331, HCXY, and SYL, which we denote as SOD1, SOD2, and SOD3 in our work, respectively. The details of the two datasets are summarized in Table 2.

**Table 1.** An example of samples and attributes in UJIIndoorLoc dataset.

| WAP1 | WAP2 | … | WAP520 | Longitude | Latitude | Floor | Building | Space ID | Relative position | User ID | Phone ID | Time stamp |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 100 | 100 | … | 100 | -7541.26 | 4864920.7782 | 2 | 1 | 106 | 2 | 2 | 23 | 1371713733 |
| 100 | 100 | … | 100 | -7536.62 | 4864934.2252 | 2 | 1 | 106 | 2 | 2 | 23 | 1371713691 |
| … | … | … | … | … | … | … | … | … | … | … | … | … |
| -62 | -59 | … | 100 | -7458.26 | 4864852.08 | 2 | 1 | 217 | 2 | 2 | 23 | 1371716146 |
| … | … | … | … | … | … | … | … | … | … | … | … | … |
| 100 | 100 | … | 100 | -7536.17 | 4864897.86 | 3 | 1 | 112 | 2 | 18 | 10 | 1371711025 |

**Table 2.** Dataset structures of UJIIndoorLoc and SODIndoorLoc.

| Dataset | Coverage ($m^2$) | Number of APs | Number of samples | Number of Buildings | Number of floors | Number of space IDs | Two coordinates |
|---|---|---|---|---|---|---|---|
| UJIIndoorLoc | 108,703 | 520 | 21,048 | 3 | 5 | 123 | Longitude/Latitude |
| SOD1 | 8,000 in total | 52 | 1,795 | 1 | 3 | 3 | ECoord/NCoord |
| SOD2 | | 347 | 12,230 | 1 | 1 | 1 | ECoord/NCoord |
| SOD3 | | 363 | 9,990 | 1 | 1 | 3 | ECoord/NCoord |


**Fig. 1** The schematic application of WiFi based indoor localization in an airport.

 

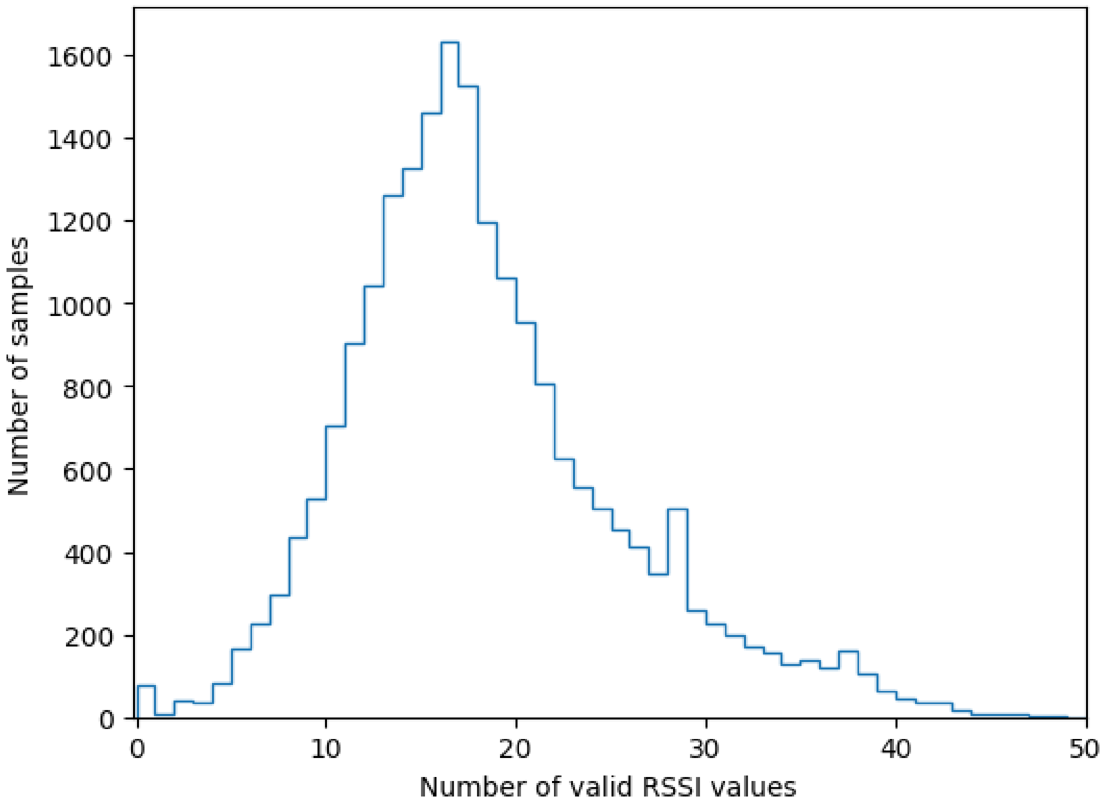


**Fig. 2** The histogram of number of samples against number of valid RSSI values in UJIIndoorLoc.

Although UJIIndoorLoc has 5 floors and 3 buildings, the first two buildings do not have a fifth floor, so there are 13 kinds of combinations of floors and buildings. The original authors split UJIIndoorLoc into a training set (19,937 samples) and a validation set (1,111 samples). The validation set includes the two coordinates of each sample but does not include space ID information, so only the training set of UJIIndoorLoc is used for the classification of space ID and floor & building in our work. However, we still used the validation set and the training set of UJIIndoorLoc for the regression of two coordinates in our work, except CNN, since CNN aimed to output both the regression and classification results simultaneously. The RSSI values in UJIIndoorLoc are represented as negative integer values ranging from -104 dBm (extremely poor signal) to 0 dBm. The authors use 100 to denote an invalid or missing AP signal. It is worth noting that the RSSI fingerprint matrix of samples in UJIIndoorLoc is quite a sparse matrix. Fig. 2 provides a histogram of the number of samples in UJIIndoorLoc against the number of valid AP signals.

We observed that nearly all samples have less than 50 valid RSSI values, although there are 520 APs in UJIIndoorLoc. In fact, there are 76 samples that do not have any valid RSSI values. However, we kept these 76 samples in UJIIndoorLoc because missing values in the dataset might carry certain meanings. The sparsity of RSSI fingerprint matrix is a typical characteristic of commonly used datasets in WiFi indoor localization studies. UJIIndoorLoc also is an imbalanced dataset regarding space ID. Fig. 3 is the histogram of space ID distribution of samples. We used space ID stratified splitting of dataset to address the imbalance. The UJIIndoorLoc dataset also provides labels of the relative position (inside/outside the space). However, we did not try to classify the relative position, because machine learning models still struggled with the classification of space ID, and it was meaningless to predict the relative position if changes were still high to misclassify the space ID.

There are 1,802 points with different locations arranged in SODIndoorLoc. 1,630 reference points and 272 testing points are included in SODIndoorLoc. The distance between two adjacent sampling points is about 1.2 m in SOD2 and SOD3 and about 0.5 m in SOD1. There are 105 pre-installed APs in the three buildings of SODIndoorLoc, with 56 being single-band and 49 being dual-band. The total coverage for SODIndoorLoc is 8000 m$^2$. For SOD2 and SOD3, the original authors provide different six csv files to download for each sub-dataset, depending on whether the readers want to use all Medium Access Controls (MACs) or only the pre-installed APs for the testing set and whether the readers want to use all 30 samples or only one averaged sample for each reference point for the training set. To be specific, we used Testing_HCXY_All.csv and Training_HCXY_All_30.csv for SOD2, and Testing_SYL_All.csv and Training_SYL_All_30.csv for SOD3. The space ID attribute in SODIndoorLoc has a different meaning from that of UJIIndoorLoc, which corresponds to the room type, *i.e.*

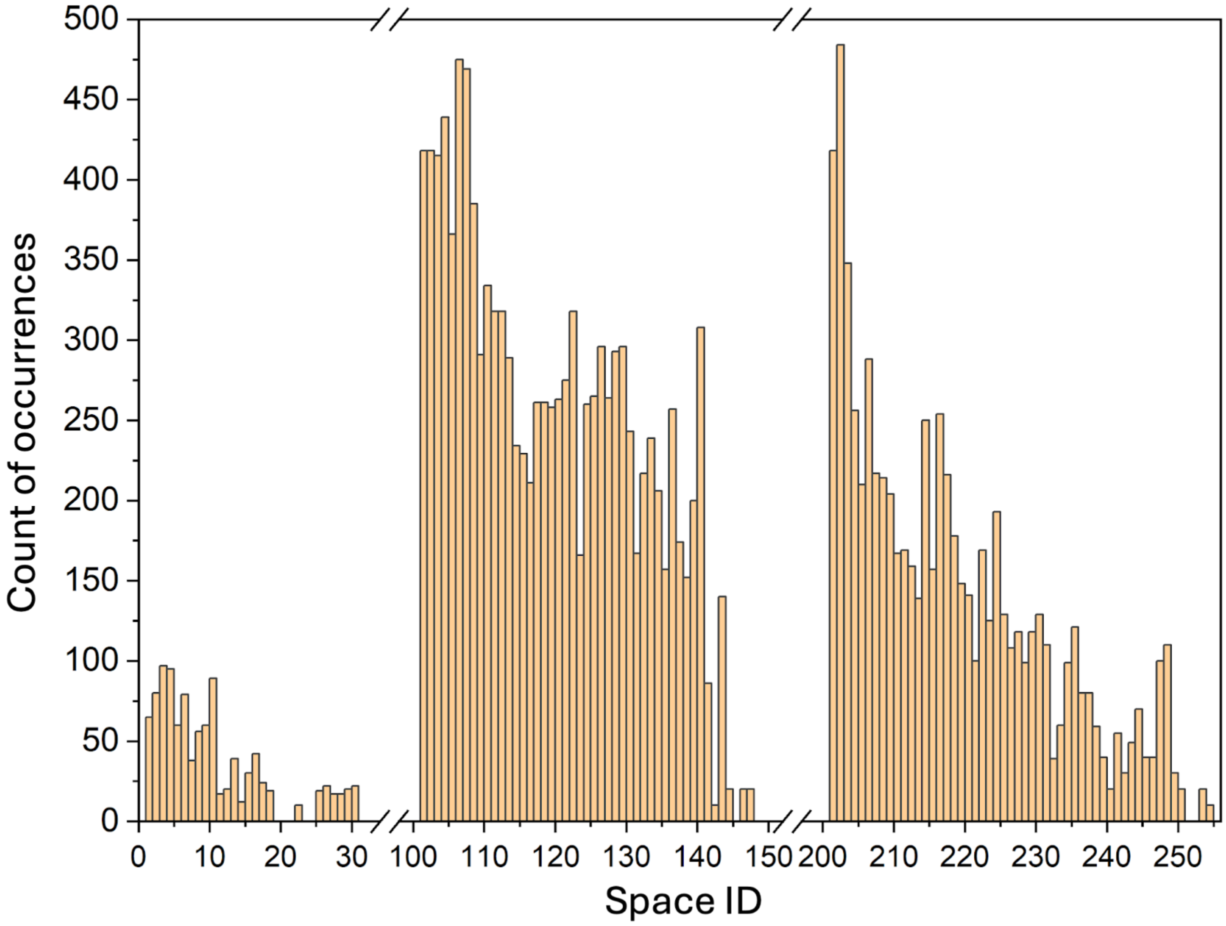


**Fig. 3** Space ID distribution of samples in UJIIndoorLoc.

corridor, office room, and meeting room, while space ID of UJIIndoorLoc corresponds to a specific room at a specific location. Since our work aims for indoor localization rather than the identification of room type, we only tried to classify the space ID for UJIIndoorLoc. Also, the classification of floor for samples in SOD1 is very easy and all models in our work could obtained 100% accuracy, so we only showed the coordinate regression results of SODIndoorLoc in our work.

### 2.2 Model development

The implementation workflow is shown in Fig. 4. In our work, we mainly used FasterKAN for WiFi indoor localization. A recently published paper by Liu *et al.*[30] introduced Kolmogorov-Arnold Networks (KAN), which claims to provide promising alternatives to Multi-Layer Perceptrons (MLPs). Rather than fixed activation functions on nodes (neurons), KAN has learnable activation functions on edges (weights). An easier implementation, efficient-KAN, was further developed to reduce memory cost.[34] After that, Li implemented FastKAN,[35] which uses Gaussian Radial Basis Function (GRBS) (Eqs. 1) to approximate the B-spline (Eqs. 2) basis, which is the bottleneck of KAN and efficient KAN. GRBS makes the forward time of FastKAN 3.33 times faster than that of efficient-KAN. Adapted from FastKAN, Delis developed FasterKAN, which uses Reflectional Switch Activation Function (RSWAF) (Eqs. 3) to approximate the B-spline basis. RSWAF is easier to calculate than GRBS while having uniform grids, and the forward time of FasterKAN is 1.5 times faster than that of FastKAN. We used FasterKAN in our work and the network structure is in Fig. 5.

$$b_i(u) = \exp\left(-\left(\frac{u-u_i}{h}\right)^2\right) \tag{1}$$

$$\varphi(x) = w(silu(x) + spline(x)) \tag{2}$$

$$b_i(u) = 1 - \left(\tanh\left(\frac{u-u_i}{h}\right)\right)^2 \tag{3}$$

### 2.3 Data preprocessing

There are 520 APs in the UJIIndoorLoc dataset, however, some APs do not contribute to the predictions of position coordinates because the RSSI values of these APs are 100 for all samples, indicating invalid or non-received RSSI.[15] After removing these useless APs, 465 APs remained in the UJIIndoorLoc dataset. Similarly, 50, 346, and 324 APs remained in the SOD1, SOD2, and SOD3 datasets respectively. We used the MinMaxScaler to transform features and fit them in the range between 0 and 1 (Eqs. 4). The MinMaxScaler speeded up FasterKAN since ensuring the input data range relieved the burden of FasterKAN to dynamically update the grid range. For other machine learning models, we used StandardScaler to transform input features. For the regression task of positioning coordinates in the UJIIndoorLoc dataset,

 

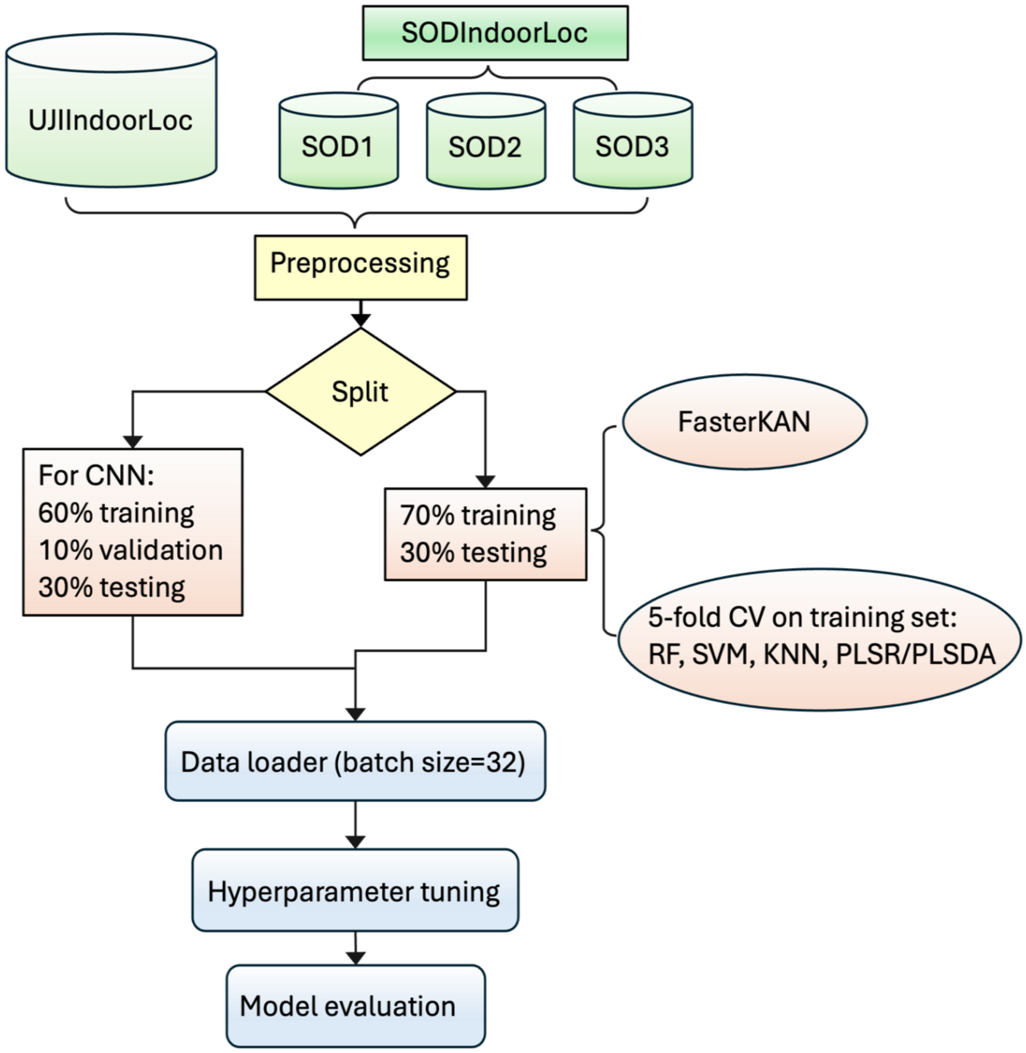


**Fig. 4** The schematic workflow of dataset processing and model development.

we deducted the minimum coordinate values from the two label coordinate values of each sample (Eqs. 5) respectively since the original label coordinate values were very large, causing numerical instability. We did not further scale the label coordinates because we wanted to keep the unit, *i.e.* meter, of the coordinates so that we could better compare our results to other studies. We used the LabelEncoder to encode the labels of floor & building as well as space ID. We split the UJIIndoorLoc and SODIndoorLoc datasets into a 70% training set and a 30% testing set respectively for all models except CNN. For CNN, these two datasets were split into a 60% training set, a 30% testing set, and a 10% validation set respectively, due to the implementation of early stopping callback to restore best weights of the model. Although SODIndoorLoc is a supplementary dataset of the UJIIndoorLoc dataset, we treated them separately in our study and model performance was evaluated on these two datasets independently. We evaluated model performance on three sub datasets of SODIndoorLoc dataset independently as well. The stratification by the space ID was used in the split for UJIIndoorLoc due to dataset imbalance. Data loaders were used with a batch size of 32 for both datasets.

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{4}$$

$$y_{transformed} = y - y_{min} \tag{5}$$

### 2.4 Hyperparameter setting

We manually tuned the hyperparameters of the FasterKAN model in a grid search style (tuning results are not shown). The hyperparameter for FasterKAN is set as {batch_size = 32; num_grids = 32; layers_list = [465, 400, 400, 2]; grid_range = [-2, 2]; exponent = 2; denominator = 0.15}. AdamW optimizer was used in training, with initial learning rate = 8e-3 and weight_decay = 8e-2. The large weight_decay value was used to better regularize the training of FasterKAN. The ReduceLROnPlateau scheduler with a factor = 0.9 was used to dynamically adjust the learning rate during training.

Apart from FasterKAN, we also implemented CNN, Support Vector Machine (SVM), Random Forest (RF), Partial Least Squares Regression (PLSR)/Partial Least Squares Discriminant Analysis (PLSDA), and k-nearest neighbors (KNN) models for performance comparisons. A simple CNN structure was used: (1) Input layer; (2) Conv1D with 64 output channels, a kernel of size 3, and the relu activation; (3) AveragePooling1D with a kernel of size 2; (4) Flatten layer;

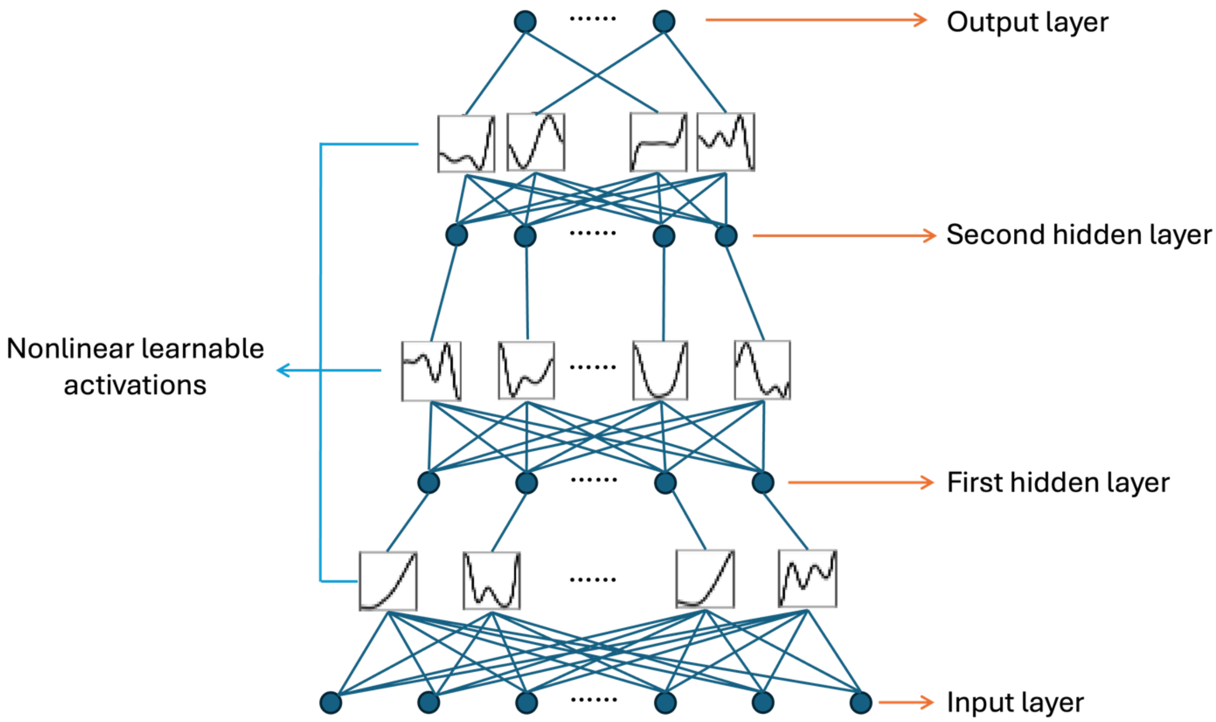


**Fig. 5** FasterKAN structure. There are 4 layers in FasterKAN: Input, first hidden, second hidden, and output layers. The number of nodes in the input and output layers depends on the number of features in the dataset. Each hidden layer contains 400 nodes. RSWAF is used as the nonlinear learnable activation function between layers.

(5) Dense with 512 nodes and the relu activation; (6) Output layer. The CNN structure for UJIIndoorLoc coordinates prediction is illustrated in Fig. 6. An early stopping callback with patience = 4 was used to restore the best weights for CNN.

For traditional machine learning models, including SVM, RF, PLS, and KNN, the grid search hyperparameter tuning scheme is in Table 3. The scatter plot of Euclidean distance positioning loss on 5-fold cross-validation (CV) for each hyperparameter set of traditional machine learning models is in Supplementary Fig. S1. The fine-tuned hyperparameters of traditional machine learning models in our work are summarized in Supplementary Table S1.

## 3. Results and discussion

To compare model performance across studies using UJIIndoorLoc dataset, a summarized view of relevant WiFi indoor localization schemes is in Table 4.

### 3.1 Indoor localization estimation

Figure 7 is the schematic diagram of model evaluation. The UJIIndoorLoc and SODIndoorLoc datasets include the two position features of longitude/latitude and ECoord/NCoord

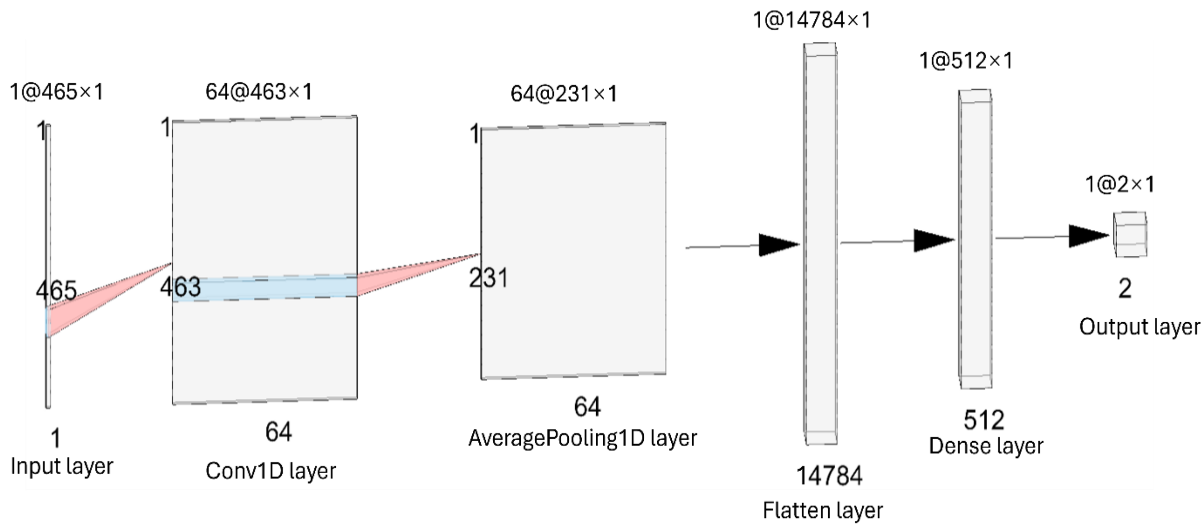


**Fig. 6** The CNN structure for UJIIndoorLoc coordinates prediction. The shape format of each layer is depth@height×width.

**Table 3.** Grid search hyperparameter tuning scheme for SVM, RF, PLSR/PLSDA, and KNN.

| Machine learning model | Hyperparameter | Values |
|---|---|---|
| PLSR/PLSDA | number of components | integers from 1 to 100 |
| KNN | number of neighbors | integers from 1 to 100 |
| SVM | gamma | 1e-3, 1e-2, 0.1, 1, 10, 100 |
| | C | 1e-3, 1e-2, 0.1, 1, 10 |
| | kernel | linear, rbf, poly, sigmoid |
| RF | number of estimators | 20, 30, 40, 50, 60, 70, 80, 90, 100 |
| | max_features | sqrt, log2 |
| | max_depth | 5, 10, 15, 20, 25, 30 |
| | min_samples_leaf | 1e-3, 5e-3, 1e-2, 5e-2, 0.1 |

**Table 4.** A summarized view of machine learning based WiFi indoor localization schemes and their performance evaluations on UJIIndoorLoc.

| Year and paper | Main machine learning model | Floor and building classification accuracy | Space ID classification accuracy | Positioning error (m) |
|---|---|---|---|---|
| Our work | FasterKAN | 99% | 71% over entire dataset | 3.56 |
| 2017[25] | DNN, SAE | 92% | - | - |
| 2018[27] | CNN | 95.41% | - | - |
| 2018[28] | GMM, RDF | - | A mean of 85% limited to each building dataset | 6.29 |
| 2019[26] | CNN, SAE | 100% for building<br>95% for floor | - | 11.78 |
| 2021[29] | CMDGRU | - | - | 6.25±0.8 |

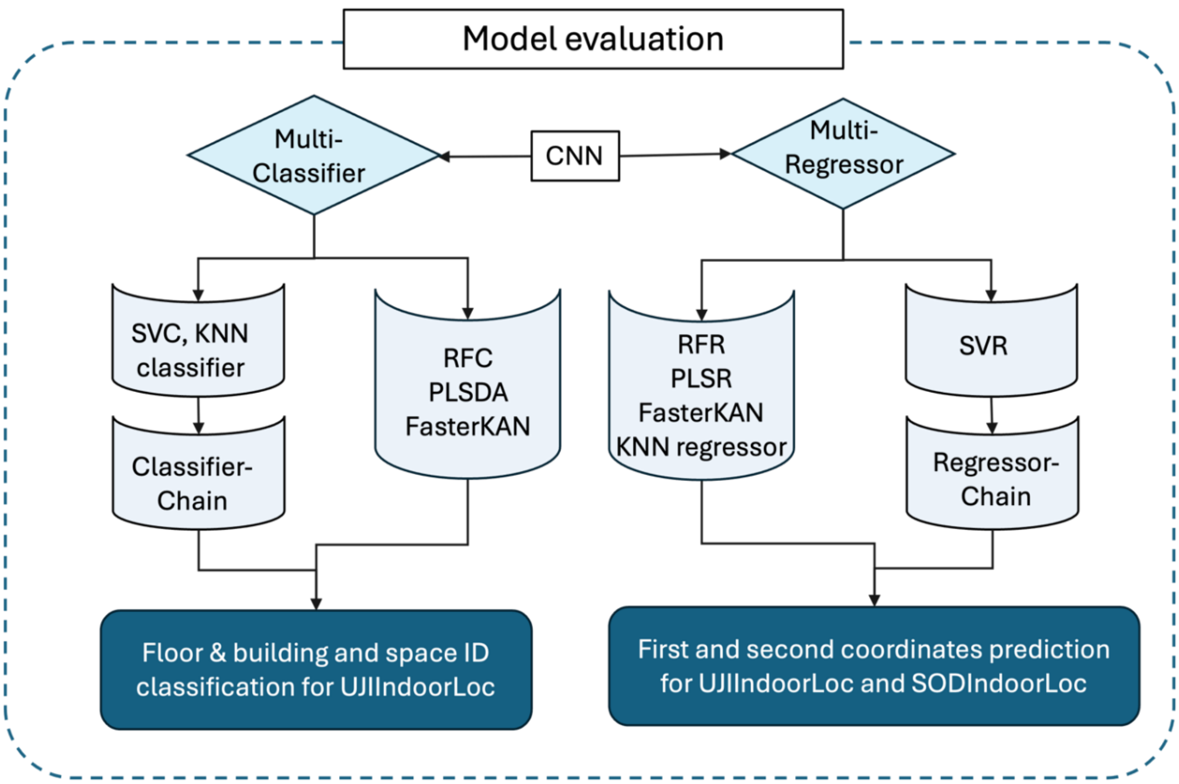


**Fig. 7** The schematic model evaluation.

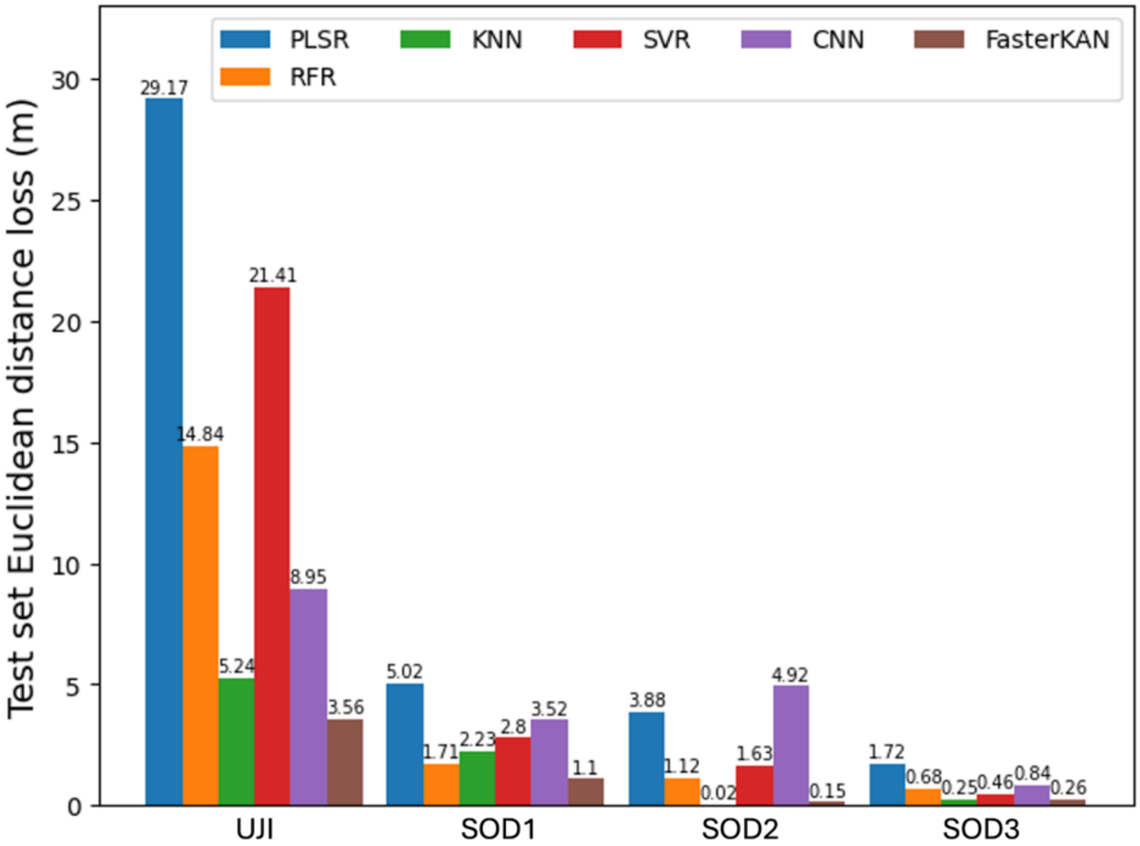


**Fig. 8** Test set mean positioning error (m) of six machine learning models on UJIIndoorLoc and SODIndoorLoc datasets.

respectively. One of the main objectives of this study is to accurately predict the coordinates of mobile device samples. It should be noted that machine learning models should output two predicted coordinates simultaneously. CNN and FasterKAN are networks implemented by TensorFlow and Pytorch and support multiple outputs by nature. Our simple CNN model outputs two predicted coordinates, categorical space ID prediction, and categorical flood and building prediction at the same time. Other models output coordinates prediction and classification results separately.

Traditional regressors, including KNN, PLSR, and RFR, support multiple regression outputs as well. However, SVR only supports single regression output. We used RegressorChain to wrap SVR so that it could output two coordinates simultaneously. RegressorChain allowed SVR to output the second coordinate, taking in inputs of all available features plus the first coordinate that SVR predicted earlier. Fig. 8 shows the results of the test set mean Euclidean distance loss (m) as in (Eqs. 6), *i.e.* positioning error, of six machine learning models on UJIIndoorLoc dataset and three sub datasets of SODIndoorLoc dataset respectively. Let x and y denote the observed first and second coordinates in UJIIndoorLoc (Longitude/Latitude) and SODIndoorLoc datasets (ECoord/NCoord) respectively. Let x' and y' denote the predicted x and y values respectively.

$$Euclidean\ distance\ loss = \sqrt{(x-x')^2+(y-y')^2} \quad (6)$$

Figure 9 provides the training and validation curves of FasterKAN and CNN. It should point out that in Fig. 9, FasterKAN used 30% validation data, i.e. testing data, and CNN only used 10% validation data. It was observed that FasterKAN performed significantly better than other models in terms of predictions of coordinates. Interestingly, KNN also showed comparatively low positioning error, which could be attributed to the way ground truth coordinates were collected in datasets. For example, at a specific location, 30 sample points were measured and included in SODIndoorLoc dataset. These 30 samples had very similar coordinates and RSSI fingerprints. It is skeptical that KNN would still provide comparatively low positioning error when applied to WiFi RSSI datasets that collect single or only a couple of sample points at a specific indoor position.

### 3.2 Space ID and floor & building classification

Another main objective of this study is to accurately classify which room, floor, and building the mobile device is located, using the WiFi RSSI fingerprint. We assigned the classification task of floor & building to machine learning models and then separately assigned the classification task of space ID while most similar studies that used UJIIndoorLoc dataset did not try to classify space ID. Other studies also developed machine learning models to classify floor and building separately, while our study aimed to classify floor & building simultaneously, by using ClassifierChain for KNN, SVC, and PLSDA. ClassifierChain worked in a similar way to

 

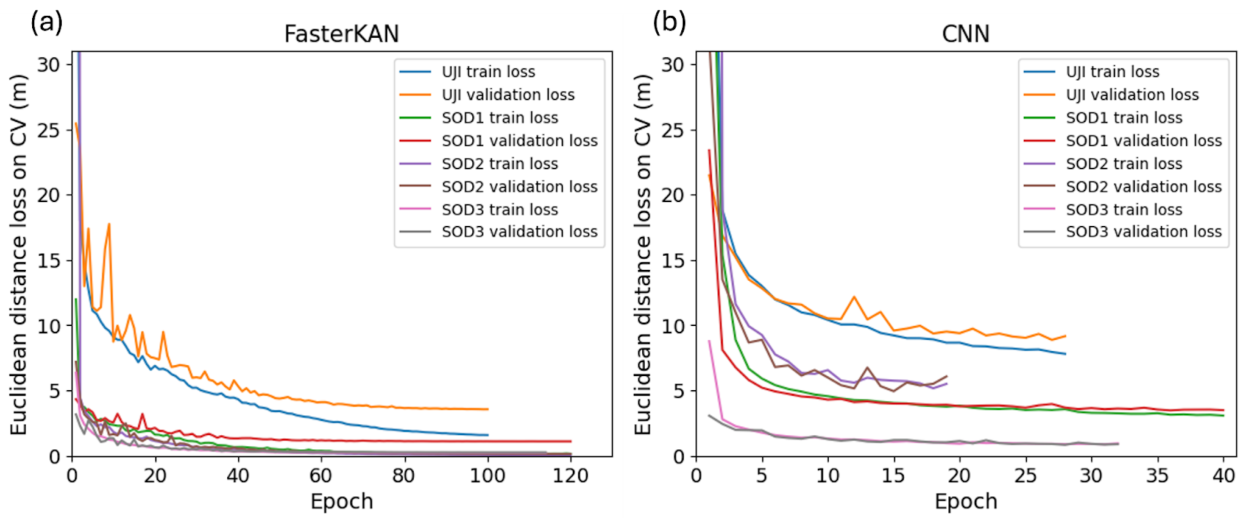


**Fig. 9** The training and validation curves of positioning error (m) of FasterKAN and CNN on UJIIndoorLoc and SODIndoorLoc datasets.

RegressorChain. Other models in our study support multiple categorical outputs by nature. It is worth noting that there was no direct implementation of PLSDA, but we created one-hot dummy testing sets to make PLSR function as PLSDA. The classification reports of weighted precision, recall, and F1 score of six machine learning on UJIIndoorLoc dataset models are summarized in Fig. 10. For SODIndoorLoc dataset, only SOD1 has different floor labels for samples and the classification is very easy. All models in our work provided 100% accuracy for classifying floor, so we only showed the model classification results on UJIIndoorLoc dataset. Supplementary Fig. S2 and Supplementary Fig. S3 are the UJIIndoorLoc floor & building and space ID classification confusion matrixes plotted in heat map for six machine learning models, respectively.

### 3.3 Model forward time and parameter efficiency

We evaluated the model forward time and total number of parameters of FasterKAN and CNN models. Under different GPU accelerating scenarios, the testing of model forward time was carried out on UJIIndoorLoc coordinate prediction and space ID classification separately. To make it more comparable between the two models, we changed the number of nodes in the Dense layer of CNN so that CNN and FasterKAN can have roughly same level of parameters. There were 751 and 848 nodes in the Dense layer of CNN for UJIIndoorLoc coordinate prediction and space ID classification respectively when testing the model forward time. Other architecture settings remained the same to Section 4.2 for CNN. We also evaluated the model forward time for traditional machine learning models in this section and their hyperparameter settings remained the same to Supplementary Table S1. We implemented the traditional machine learning models by the sklearn package, which does not aim to use GPU accelerating. We summarized the results of the total number of parameters and the average model forward time (μs) over many trials in Table 5. In each trial, we recorded the total forward time of 5982 samples in the test set and calculated the average forward time (μs) for each sample. For the forward time evaluation on GPU accelerating, we tested T4

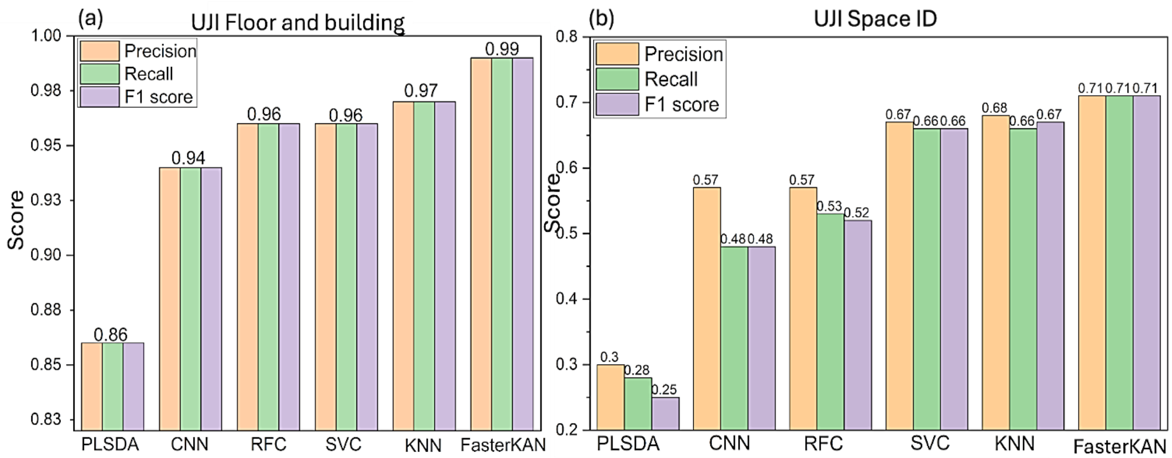


**Fig. 10** Test set classification report of precision, recall, and F1 score of six machine learning models on UJIIndoorLoc dataset (a) Floor and building classification results, (b) Space ID classification results.

**Table 5.** The total number of parameters and average forward time (μs) for each model.

| Machine learning model | Total number of parameters | Average forward time (μs) | Number of trials | GPU/Memory |
|---|---|---|---|---|
| FasterKAN regressor | 11,100,226 | 13.51 | 1000 | T4/15GB |
| FasterKAN regressor | 11,100,226 | 2.81 | 1000 | A100/41GB |
| FasterKAN regressor | 11,100,226 | 3.40 | 1000 | L40/46GB |
| FasterKAN regressor | 11,100,226 | 130.49 | 1000 | - |
| CNN regressor | 11,105,295 | 99.12 | 1000 | T4/15GB |
| CNN regressor | 11,105,295 | 23.02 | 1000 | A100/41GB |
| CNN regressor | 11,105,295 | 22.16 | 1000 | L40/46GB |
| CNN regressor | 11,105,295 | 353.93 | 1000 | - |
| FasterKAN classifier | 12,649,026 | 13.89 | 1000 | T4/15GB |
| FasterKAN classifier | 12,649,026 | 3.00 | 1000 | A100/41GB |
| FasterKAN classifier | 12,649,026 | 3.42 | 1000 | L40/46GB |
| FasterKAN classifier | 12,649,026 | 183.50 | 1000 | - |
| CNN classifier | 12,642,363 | 98.40 | 1000 | T4/15GB |
| CNN classifier | 12,642,363 | 23.59 | 1000 | A100/41GB |
| CNN classifier | 12,642,363 | 24.79 | 1000 | L40/46GB |
| CNN classifier | 12,642,363 | 399.58 | 1000 | - |
| SVR | - | 15930 | 10 | - |
| SVC | - | 8112 | 10 | - |
| RFR | - | 12.43 | 1000 | - |
| RFC | - | 83.82 | 1000 | - |
| PLSR | - | 3.04 | 1000 | - |
| PLSDA | - | 9.09 | 1000 | - |
| KNN regressor | - | 380.90 | 100 | - |
| KNN classifier | - | 806.61 | 100 | - |

accelerating on Google Colaboratory,[36] and tested A100 and L40 accelerating on Storrs High Performance Computing (HPC) center. Since L40 prefers single-precision, we used TensorFloat-32 for the matrix multiplication. For CPU only testing, we used the Epyc128 architecture on Storrs HPC. SVR needed a RegressorChain wrapper to output two coordinates simultaneously in the forward time testing. Ideally, the model forward time would be halved if SVR is to output a single value. However, SVM forward time was still too high and not comparable to other models. The results indicated that when using GPU accelerating and CPU only, the model forward time of FasterKAN was about 8.2 and 3.7 times shorter than that of CNN model respectively, with roughly same number of total parameters.

## 4. Conclusion

In this study, we applied the cutting-edge FasterKAN to two widely used multi-floor and multi-building WiFi RSSI fingerprint datasets, i.e. UJIIndoorLoc and SODIndoorLoc, for indoor localization. FasterKAN's position errors (m) on test set were 3.56, 1.1, 0.15, and 0.26 for UJIIndoorLoc, SOD1, SOD2, and SOD3 datasets, respectively. FasterKAN achieved 99% and 71% classification accuracy for floor & building and space ID on UJIIndoorLoc dataset respectively. It outperformed traditional machine learning models developed in our work and other deep learning models developed in similar studies. On the other hand, KAN is faster and more parameter-efficient compared to MLP based networks. The forward time of FasterKAN is much shorter than original KAN implementation, addressing the short latency needs in WiFi based indoor localization. The model forward time of FasterKAN was about 8 times shorter than that of CNN model with roughly same number of total parameters. Although latency might consider WiFi handover delay as well, this promising result at least showed great potential of FasterKAN-integrated system to shorten the backend processing time of indoor localization.

The advancements made by FasterKAN in WiFi indoor localization can significantly benefit a variety of real-world applications. For instance, in large and complex environments such as airports, shopping malls, and hospitals, accurate and efficient indoor localization can enhance navigation and wayfinding for visitors, improving their overall experience. Additionally, FasterKAN's robust and scalable solution can be utilized in smart buildings and offices to optimize space utilization, manage resources, and enhance security through precise tracking of assets and personnel. In industrial settings, such as warehouses and factories, improved localization accuracy and reduced latency can streamline operations, facilitate inventory management, and ensure worker safety. Overall, the application of FasterKAN in these scenarios underscores its potential to revolutionize indoor localization across various sectors, making environments smarter, safer, and more efficient.

 

For future work, we plan to work on a benchmark comparison in terms of qualitative parameter-efficiency, forward time, and accuracy in handling WiFi RSSI fingerprint datasets between FasterKAN and other MLP or CNN based deep learning models as well as other variants of KAN implementations. Apart from that, we plan to potentially design a deep network based on FasterKAN that specifically targets WiFi RSSI fingerprint datasets and take advantage of the typical sparsity of such datasets.

## Conflict of Interest

There is no conflict of interest.

## Supporting Information

Applicable.

## References


[1] S. He, S.-H G. Chan, Wi-Fi fingerprint-based indoor positioning: recent advances and comparisons, *IEEE Communications Surveys & Tutorials*, 2016, **18**, 466-490, doi: 10.1109/COMST.2015.2464084.

[2] J. Kunhoth, A. Karkar, S. Al-Maadeed, A. Al-Ali, Indoor positioning and wayfinding systems: a survey, *Human-Centric Computing and Information Sciences*, 2020, **10**, 18, doi: 10.1186/s13673-020-00222-0.

[3] J. Luo, L. Fan, H. Li, Indoor positioning systems based on visible light communication: state of the art, *IEEE Communications Surveys & Tutorials*, 2017, **19**, 2871-2893, doi: 10.1109/COMST.2017.2743228.

[4] H. Xu, Y. Ding, P. Li, R. Wang, Y. Li, An RFID indoor positioning algorithm based on Bayesian probability and K-nearest neighbor, *Sensors*, 2017, **17**, 1806, doi: 10.3390/s17081806.

[5] M. Ridolfi, S. Vandermeeren, J. Defraye, H. Steendam, J. Gerlo, D. De Clercq, J. Hoebeke, E. De Poorter, Experimental evaluation of UWB indoor positioning for sport postures, *Sensors*, 2018, **18**, 168, doi: 10.3390/s18010168.

[6] S. Shang, L. Wang, Overview of WiFi fingerprinting-based indoor positioning, *IET Communications*, 2022, **16**, 725-733, doi: 10.1049/cmu2.12386.

[7] S. Yiu, M. Dashti, H. Claussen, F. Perez-Cruz, Wireless RSSI fingerprinting localization, *Signal Processing*, 2017, **131**, 235-244, doi: 10.1016/j.sigpro.2016.07.005.

[8] S. G. Obreja, A. Vulpe, Evaluation of an indoor localization solution based on bluetooth low energy beacons, 2020 13th International Conference on Communications (COMM). Bucharest, Romania. IEEE, 2020.

[9] F. Daniş, A. Cemgil, Model-based localization and tracking using bluetooth low-energy beacons, *Sensors*, 2017, **17**, 2484, doi: 10.3390/s17112484.

[10] A. Musa, A.-A A. Dabo, A review of RFID in supply chain management: 2000–2015, *Global Journal of Flexible Systems Management*, 2016, **17**, 189-228, doi: 10.1007/s40171-016-0136-2.

[11] J. Liu, R. Jain, Survey of wireless based indoor localization technologies, Department of Science & Engineering, Washington University, 2014.

[12] 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, 2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN). Busan, Korea (South). IEEE, 2014.

[13] J. Bi, Y. Wang, B. Yu, H. Cao, T. Shi, L. Huang, Supplementary open dataset for WiFi indoor localization based on received signal strength, *Satellite Navigation*, 2022, **3**, 25, doi: 10.1186/s43020-022-00086-y.

[14] G. Sithole, S. Zlatanova, Position, location, place and area: an indoor perspective, *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2016, **III-4**, 89-96, doi: 10.5194/isprsannals-iii-4-89-2016.

[15] N. Singh, S. Choe, R. Punmiya, Machine learning based indoor localization using Wi-Fi RSSI fingerprints: an overview, *IEEE Access*, 1083, **9**, 127150-127174, doi: 10.1109/ACCESS.2021.3111083.

[16] Z. Yang, C. Wu, Y. Liu, Locating in fingerprint space: wireless indoor localization with little human intervention, Proceedings of the 18th annual international conference on Mobile computing and networking, *Istanbul Turkey*, *ACM*, 2012, 269-280, doi: 10.1145/2348543.2348578.

[17] D. A. Tran, C. Pham, Fast and accurate indoor localization based on spatially hierarchical classification, 2014 IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems. Philadelphia, PA, USA. IEEE, 2014.

[18] W. Zhang, K. Liu, W. Zhang, Y. Zhang, J. Gu, Deep Neural Networks for wireless localization in indoor and outdoor environments, *Neurocomputing*, 2016, **194**, 279-287, doi: 10.1016/j.neucom.2016.02.055.

[19] K. A. Nguyen, A performance guaranteed indoor positioning system using conformal prediction and the WiFi signal strength, *Journal of Information and Telecommunication*, 2017, **1**, 41-65, doi: 10.1080/24751839.2017.1295659.

[20] M. Mohammadi, A. Al-Fuqaha, M. Guizani, J.-S. Oh, Semisupervised deep reinforcement learning in support of IoT and smart city services, *IEEE Internet of Things Journal*, 2018, **5**, 624-635, doi: 10.1109/JIOT.2017.2712560.

[21] Y. Li, X. Hu, Y. Zhuang, Z. Gao, P. Zhang, N. El-Sheimy, Deep reinforcement learning (DRL): another perspective for unsupervised wireless localization, *IEEE Internet of Things Journal*, 2020, **7**, 6279-6287, doi: 10.1109/JIOT.2019.2957778.

[22] F. Dou, J. Lu, Z. Wang, X. Xiao, J. Bi, C.-H. Huang, Top-down indoor localization with Wi-Fi fingerprints using deep Q-network, 2018 IEEE 15th International Conference on Mobile Ad Hoc and Sensor Systems (MASS). Chengdu, China. IEEE, 2018.

[23] X. Lin, J. Gan, C. Jiang, S. Xue, Y. Liang, Wi-Fi-based indoor localization and navigation: a robot-aided hybrid deep learning approach, *Sensors*, 2023, **23**, 6320, doi: 10.3390/s23146320.

[24] H. Ye, J. Peng, Robot indoor positioning and navigation based on improved WiFi location fingerprint positioning algorithm, *Wireless Communications and Mobile Computing*, 2022, **2022**, 8274455, doi: 10.1155/2022/8274455.
[25] M. Nowicki, J. Wietrzykowski, Low-effort place recognition with WiFi fingerprints using deep learning. R. Szewczyk, C. Zieliński, M. Kaliczyńska, eds. Advances in Intelligent Systems and Computing. Cham: Springer International Publishing, 2017.
[26] 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*, 2019, **7**, 110698-110709, doi: 10.1109/ACCESS.2019.2933921.
[27] J.-W. Jang, S.-N. Hong, Indoor localization with WiFi fingerprinting using convolutional neural network, 2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN). Prague, Czech Republic. IEEE, 2018.
[28] B. A. Akram, A. H. Akbar, O. Shafiq, HybLoc: hybrid indoor Wi-Fi localization using soft clustering-based random decision forest ensembles, *IEEE Access*, 2018, **6**, 38251-38272, doi: 10.1109/ACCESS.2018.2852658.
[29] W. Qian, F. Lauri, F. Gechter, Supervised and semi-supervised deep probabilistic models for indoor positioning problems, *Neurocomputing*, 2021, **435**, 228-238, doi: 10.1016/j.neucom.2020.12.131.
[30] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, T. Y. Hou, M. Tegmark, KAN: kolmogorov-arnold networks, 2024: arXiv: 2404.19756, doi: 10.48550/arXiv.2404.19756.
[31] Y. Hou, D. Zhang, A comprehensive survey on Kolmogorov Arnold networks (KAN), 2024: arXiv: 2407.11075, doi: 10.48550/arXiv.2407.11075.
[32] S. Karakaş, A novel approach to predicting liquefaction-induced settlements using Kolmogorov–Arnold Networks (KANs), *Discover Geoscience*, 2024, **2**, 77, doi: 10.1007/s44288-024-00082-6.
[33] Z. Liu, P. Ma, Y. Wang, W. Matusik, M. Tegmark, KAN 2.0: kolmogorov-arnold networks meet science, 2024: arXiv: 2408.10205, doi:10.48550/arXiv.2408.10205.
[34] Chen Z, Zhang X. Lss-skan: Efficient kolmogorov-arnold networks based on single-parameterized function, arxiv preprint arxiv:2410.14951, 2024, doi:10.48550/arXiv.2410.14951.
[35] Z. Li, Kolmogorov-arnold networks are radial basis function networks, 2024: arXiv: 2405.06721., doi: 10.48550/arXiv.2405.06721.
[36] E. Bisong, Google colaboratory, Building machine learning and deep learning models on google cloud platform, Berkeley, CA: Apress, 2019.



 