Transactions on Energy Systems and Engineering Applications

Article

HyTra: Hyperclass Transformer for WiFi Fingerprinting-based Indoor Localization

Muneeb Nasir[1], Kiara Esguerra[1], Ibrahima Faye[1], Tong Boon Tang[1], Mazlaini Yahya[2], Afidalina Tumian[2] and Eric Tatt Wei Ho[1,*]

1 Department of Electrical & Electronics Engineering, Universiti Teknologi PETRONAS, Perak, Malaysia.
2 Petroliam Nasional Berhad (PETRONAS), Malaysia.
* Correspondence: hotattwei@utp.edu.my

Received: 23 July 2023; Accepted: 1 November 2023; Published: 13 February 2024

Abstract: The emerging demand for a variety of novel Location-based Services (LBS) by consumers and industrial users is driven by the rapid and extensive proliferation of mobile smart devices. Sensors embedded in smart devices or machines provide wireless connectivity and Global Positioning System (GPS) capability, and are co-utilized to acquire location-linked data which are algorithmically transformed into reliable and accurate location estimates. GPS is a mature and reliable technology for outdoor localization but indoor localization in a complex multi-storey building environment remains challenging due to fluctuations in wireless signal strength arising from multipath fading. Location-linked data from wireless access points (WAPs) such as received signal strength (RSS) are acquired as numerical sequences. By conceptualizing a fixed order sequence of WAP measurements as a sentence where the RSS from each WAP are words, we may leverage on recent advances in artificial intelligence for natural language processing (NLP) to enhance localization accuracy and improve robustness against signal fluctuations. We propose the hyper-class Transformer (HyTra), an encoder-only Transformer neural network which learns the relative positions of wireless access points (WAPs) through multiple learnable embeddings. We propose a second network, HyTra-HF, which improves upon HyTra by applying a hierarchical relationship between location classes. We test our proposed networks on public and private datasets varying in sizes. HyTra-HF outperforms existing deep learning solutions by obtaining 96.7% accuracy for the floor classification task on the UJIIndoorloc dataset. HyTra-HF is amenable to deep model compression and achieves accuracy of 95.95% with over ten-fold reduction in model size using Sparsity Aware Orthogonal (SAO) initialization and has the best-in-class accuracy for the sparse model.

How to cite this article: Nasir, Muneeb; Esguerra, Kiara; Faye, Ibrahima; Tang, Tong Boon; Yahya, Mazlaini; Tumian, Afidalina; Ho, Eric Tatt Wei. HyTra: Hyperclass Transformer for WiFi Fingerprinting-based Indoor Localization. *Transactions on Energy Systems and Engineering Applications*, 5(1): 542, 2024. DOI:10.32397/tesea.vol5.n1.542

1. Introduction

Location and proximity-based services have been utilized in many different environments and use cases, from industrial warehouses to airports and hospitals; given the nature of their application (e.g., monitoring and navigation), precise position information is often required [1]. Approximately 70% of smartphone usage and 80% of data transmission occurs indoors [2]; given that several market reports anticipate significant growth, the global localization market is expected to advance to $402 billion by 2031 [3]. The increasing demand for Internet of Things (IoT) capable machines within the industrial sector and the proliferation of smart devices both aids existing services and incentivizes the creation of new location-based services. Moreover, the recent Covid-19 pandemic warranted contact tracing in closed spaces. Localization with highly accurate position information was crucial to track and minimize the spread of the virus.

Localization is the process of identifying the position of a given device within a given operating environment, which can be either outdoor or indoor. The global navigation satellite system (GNSS) is the leading standard for outdoor localization. The main GNSS constellations and their operators are Global Positioning System (GPS) by the USA, Globalnaya Navigatsionnaya Sputnikovaya Sistema (GLONASS) by Russia, Galileo by Europe and Beidou by China, with GPS being the more popular system. These individual satellite systems are well-established within outdoor localization and deliver accurate readings outdoors. However, in closed indoor environments, due to non-line of sight conditions, walls, ceilings, and furniture obstruct and further weaken (by absorbing) the already low-powered signal, making it difficult to accurately determine the device or user location. Thus, a satellite-based system is not a viable solution for indoor localization. These satellite systems are deemed ineffective due to the substantial amount of signal attenuation and interference experienced by the signal as it propagates through a building, resulting in inaccurate indoor position estimates.

Indoor localization methods may be distinguished by the operating principle of the algorithm; if it utilizes a time, geometric or fingerprinting-based approach. Fingerprinting methods are the most popular approach, particularly WiFi fingerprinting based on received signal strength (RSS) [4]. This is no surprise due to the ubiquity of WiFi connectivity indoors. Ease and cost of implementation also play a major factor as existing WiFi infrastructure can be availed. Although WiFi fingerprinting using Channel State Information (CSI) is more precise than its RSS counterpart, an additional network card is required to obtain CSI data rendering it a less attractive solution.

Regardless of its popularity, RSS-based WiFi fingerprinting faces difficulty in precise localization due to the fluctuating noisy RSS data caused by multipath fading as a result of the dynamic operating environment; a lack of large (100 thousand plus samples) multi-building multi-floor RSS datasets further exacerbates this issue. Traditional Machine Learning (ML) methods have been employed to tackle this issue; classification approaches like Random Forests, Support Vector Machines, and K-Nearest Neighbours work well on smaller datasets but require extensive feature engineering and parameter tuning. Furthermore, the performance of traditional ML methods will not scale with larger, more complex datasets. Deep Learning (DL) methods can capture complex relationships amongst features without explicit feature engineering, and their performance scales positively with increments in data.

Methods such as Deep Neural Networks (DNN), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN) and Long-Short Term Memory (LSTM) have been extensively tested by the DL community [5]. The Transformer network [6] offers several advantages over other sequence-processing neural networks like Recurrent Neural Networks (RNN) [7] and Long-Short Term Memory (LSTM) [8] and has been effectively deployed in applications for natural language processing [9, 10, 11], computer vision [12], and tested for indoor localization using CSI [13, 14]. The Transformer uses an attention mechanism to capture pairwise relationships while effectively learning long-range dependencies in the input. Neural networks scale in size and computational cost with problem complexity, a trend that is evident in Natural Language Processing (NLP) tasks [15]. Large, high-accuracy models are difficult to deploy on Internet-of-Things (IoT) devices. As such there is resurgent interest in model compression techniques such as weight pruning [16, 17] to reduce the size of large models without compromising accuracy [18, 19, 20].

In this study, we develop a novel RSS-based WiFi fingerprinting solution for indoor localization utilizing the Transformer neural network to precisely identify indoor locations from WiFi RSS measurements obtained from WAPs. In our work, we propose novel adaptations to and apply the Transformer network to RSS-based WiFi fingerprinting. The primary contributions of our paper in comparison to existing methods are as follows:

- We present two novel solutions for indoor localization utilizing the Transformer Encoder network for RSS-based WiFi fingerprinting. The proposed solutions are purpose-built with multiple classification heads for a complex hierarchical environment to deliver consistent and accurate results.
- We modify the Transformer network to use learnable positional embeddings (HyTra). We also imposed additional connectivity between the attention-filtered values (AFV) to promote hierarchical learning between classes (HyTra-HF). We observe an interesting feature of indoor localization classification. There appears to be an inverse relationship between classification accuracy and class locality. We posit that reusing features from coarse class (i.e. building) may improve the classification accuracy of fine class (i.e. room).
- We demonstrate that the HyTra-HF architecture is amenable to significant model compression with minimal degradation in accuracy. HyTra-HF was compressed by more than a ten-fold factor and achieved equivalent accuracy but at 60% of the trainable parameters compared to the state-of-art (CNNLoc) technique on UJIIndoorloc dataset with.
- We evaluate on one public (UJI [16]) and two private datasets (SPOT and UTP), each differing in size, demonstrating HyTra's ability to scale positively with larger datasets based on its performance across all datasets.

2. Related Works

In this section, we give a general outline of the different methods employed for indoor localization and discuss existing RSS-based WiFi printing solutions utilizing deep learning methods. Indoor localization covers a broad spectrum of techniques, each requiring different data types to perform localization. We have categorized the different indoor localization methods based on the kind of data used:

- Time Approach: Time of Flight (ToF), Time Difference of Arrival (TDoA) and Return Time of Flight (RTOF) make use of the time it took for the signal to propagate to the receiver, the time intervals between each signal reception and the signal propagation round trip time, respectively. These methods, although accurate, are affected by clock synchronization, sampling rate, and signal bandwidth; ToF also requires line-of-sight for accurate performance [1, 21].
- Geometric Approach: Angle of Arrival (AoA) and Phase of Arrival (PoA) rely on angle and phase estimation using antenna arrays to calculate the difference in arrival time and the distance between transmitter and receiver, respectively. These methods can deliver high accuracy but suffer degradation from non-line-of-sight, faded multipath signals and require complex hardware and algorithms to undo [1, 21].
- Fingerprinting: captures an array of received signal strength (RSS) or channel state information (CSI) measurements at every reference point to build a collection of signals, which are used to compare with real-time measurements to pinpoint the user's location. This technique of comparing signals is effective as each location would have a unique fingerprint arising from the complex signal propagation

within the indoor environment. CSI offers better accuracy but is prone to noise and multipath fading; however, RSS is more easily obtained and cost-effective when compared to CSI, which requires an off-the-shelf network interface card (NIC) [1, 14, 21].

WiFi fingerprinting exploits machine learning and deep learning techniques as its main algorithm [22, 23]. A database of WiFi signals is constructed by collecting the WiFi RSS at selected reference locations within the targeted indoor environment, and then a localization algorithm is trained on the collected data. During the inference stage, the trained algorithm is fed an array of newly collected RSS values from the user. The trained model predicts a location based on the similarity of the user-sent RSS array and the stored RSS data. This process is viewed as two separate phases: the offline and online phases, respectively. Figure 1 illustrates the two phases of WiFi fingerprinting.

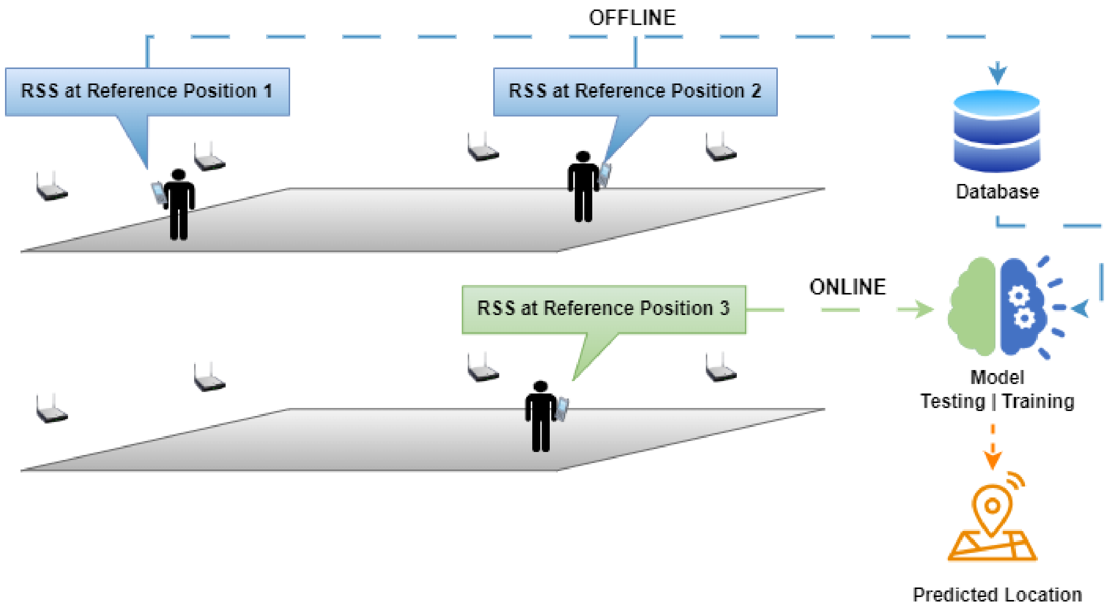

OFFLINE
RSS at Reference Position 1
RSS at Reference Position 2
Database
ONLINE
RSS at Reference Position 3
Model
Testing | Training
Predicted Location

Figure 1. Standard RSS-based indoor WiFi fingerprinting method, with a trained machine learning or deep learning model as its prediction system.

Various classification and regression algorithms have been experimented with for WiFi fingerprinting-based indoor localization, such as k-Nearest Neighbour (KNN) [24], Support Vector Machine (SVM) [25], Gradient Boosting [26], Multilayer Perceptron (MLP) [27], Convolutional Neural Network (CNN) [28, 29], Recurrent Neural Network (RNN) [30, 31], and Long Short-Term Memory (LSTM) [31, 32] have been tested. These methods fall under machine learning and deep learning (a specialized machine learning subfield). Machine learning sits at the intersection of computer science and statistics; it leverages statistical learning to extract patterns and make predictions from structured labelled data. It can be applied to unstructured data but requires some preprocessing. In contrast, deep learning is a set of techniques enabling systems to learn complex behaviours by observing unstructured data, like text, images, and audio. Deep learning requires less human manipulation in the algorithm design stage because neural networks learn to extract essential features during training but require significantly more data than machine learning methods [33].

Stacked autoencoders (SAE) are commonly employed in deep learning-based RSS-based indoor localization. An autoencoder comprises an encoder and a decoder; the encoder learns a compressed representation of the RSS input, and the decoder reconstructs the original input from the compact representation. The stacked denoising autoencoder originally proposed by [34] allows dimension reduction and extraction of key features from the input, such as the sparse, noisy RSS input. Nowicki and Wietrzykowski [35] pre-train the SAE to obtain an encoder effective at summarizing the RSS input; a deep neural network (DNN) is trained on top of the encoder to classify buildings and floors. This method achieves 92% for the building and floor prediction on the UJIIndoorLoc dataset (UJI). While [35] makes a combined prediction for building and floor, Kim et al. [36] propose to leverage the hierarchical nature of a multi-building, multi-floor complex to generate a label vector made of independent one-hot encodings of each identifier. Employing a similar network architecture and training strategy, Scalable DNNs achieves 99.5% and 91.26% accuracy on the UJI dataset using one-hot encoded hierarchical labels. Song et al. propose another SAE-based network, CNNLoc [29] but they attach a 1-dimensional convolutional neural network (CNN) to the pretrained SAE. Parameter sharing and sparse, local connections make CNNs less susceptible to overfitting. We believe that for these reasons, CNNLoc achieves an improved 100% and 96% accuracy for building and floor classification, respectively. Qin et al. employ a convolution-denoising autoencoder (CDAE) followed by another CNN for classification—dubbed Ccpos [37]. Original and noise-induced (Gaussian white noise) RSS data is fed to the CDAE to help reduce overfitting. Ccpos achieves accurate location estimates on the Alcala Tutorial 2017 dataset [38] but not on the UJI dataset compared to existing methods, with an average positioning error of 1.05 meters and 12.40 meters on the respective datasets.

Laska and Blakenbach [39] introduced a custom label encoding scheme, output layer and loss function. Their proposed model, DeepLocBox, estimates a region in which the user is located via bounding boxes. DeepLocBox achieves a best score of 99.64% and 92.62% accuracy for building and floor classification on the UJI dataset out of 10 trials. In [40], Laska and Blakenbach propose multi-cell encoding learning to solve multi-task learning problems using a single forward pass network. Using a CNN backbone, the proposed network simultaneously classifies grid cells and does in-cell regression to achieve a combined accuracy of 95.3% for building and floor classification and a mean positioning error of 7.18 meters, respectively. Recurrent neural networks (RNN) have been tested for RSS-based WiFi fingerprinting [41] and have shown to be a competitive solution to their DNN counterparts. The hierarchical RNN [41] uses the SAE to denoise and reduce the dimensionality of the RSS input. Due to the sequential nature of RNNs, hierarchical RNN effectively leverages the hierarchical underpinnings of multi-building multi-floor data to obtain 100% and 95.24% accuracy for building and floor classification on the UJI dataset

3. Methodology

This section covers the datasets used to evaluate the efficacy of the proposed networks, then discusses the preprocessing strategy, presents the proposed models, and explains their essential components. Lastly, we describe the training process and the parameters involved in obtaining the results discussed in the following section.

3.1. Datasets

3.1.1. UJIIndoorLoc (UJI)

UJIIndoorLoc [16] is one of the largest publicly available and the most widely referenced dataset within the indoor localization literature, easily accessible from the UC Irvine machine learning repository. The UJI dataset was compiled in 2013 at the Jaume I University, Castelló de la Plana, Valencian Community, Spain and is partitioned into a training and validation set comprising 19,937 and 1,111 records, respectively. Twenty (20) users on twenty-five (25) different Android devices took measurements across three (3) buildings, each with four (4) floors on average, spanning a space of 110,000 m^2. The training and testing (addressed as validation in the UJI paper) sets were generated four months apart to ensure data independence; however, in doing so, we assume a significant portion of the testing locations do not intentionally overlap with the training locations. We map all the reference points using the provided latitude and longitude values and floor IDs in Figure 2.

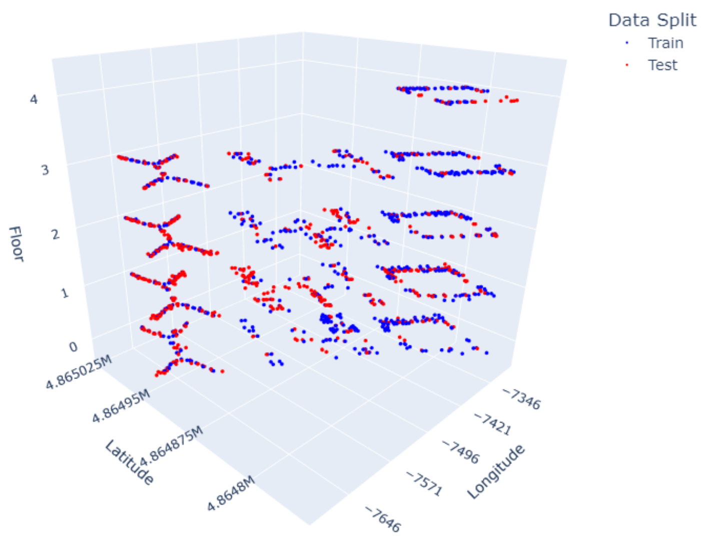
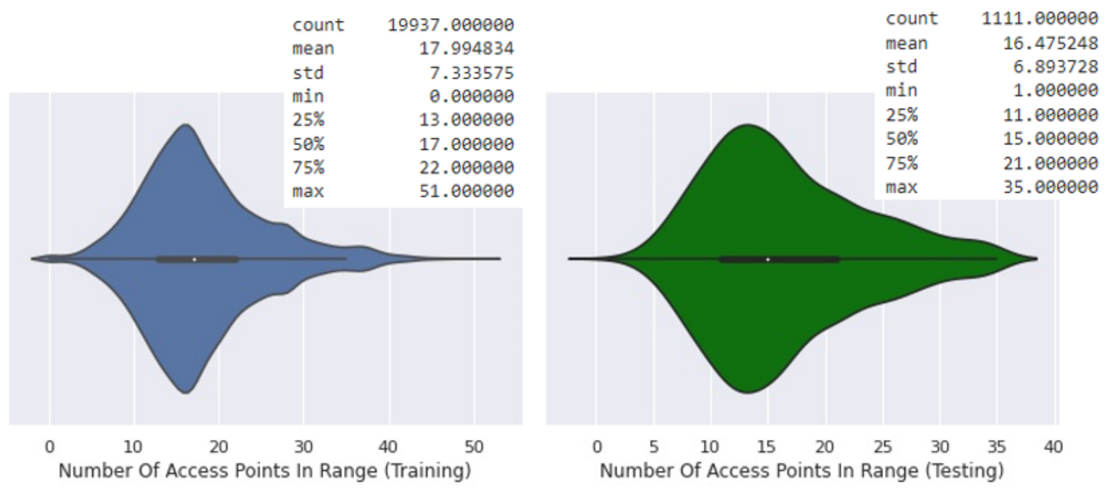


Figure 2. Training (blue) and testing (red) locations in UJIIndoorLoc; mapped by Longitude, Latitude and Floor.

The dataset contains RSS readings from 520 unique wireless access points ranging from -104 dBm (weakest detected signal) to 0 dBm (strongest detected signal). The 520 unique wireless access points are the summation of all the different access points encountered during the data collection process (training and testing). For any given measurement, the number of available access points is much less than the total number of access points–as shown in Figure 3. The authors [16] use positive 100 dBm to indicate the absence of an access point (out-of-range values).



```
count    19937.000000
mean        17.994834
std          7.333575
min          0.000000
25%         13.000000
50%         17.000000
75%         22.000000
max         51.000000
```

```
count    1111.000000
mean       16.475248
std         6.893728
min         1.000000
25%        11.000000
50%        15.000000
75%        21.000000
max        35.000000
```

Figure 3. Distribution and statistics of the number of access points within range for both the training and testing data split for the UJI dataset.

3.1.2. SPOT

The SPOT dataset is our larger private dataset curated by PETRONAS RESEARCH SDN BHD (PRSB). It roughly consists of two hundred thousand (200K) samples in total, split into training, validation and testing with a ratio of 80:10:10, respectively. RSS data was collected across five buildings with four floors on average and around two hundred unique rooms spanning the five buildings. The collected data was stored and organized in an approach similar to [16]. Data was collected continuously over a period of two months from multiple users using a purpose-built in-house IoT device.

3.1.3. UTP

A smaller dataset was created to test our proposed solution for real-time inferencing. Hence, samples were first collected at the multi-storey academic blocks of Universiti Teknologi Petronas. Roughly a thousand samples were collected and split into training, validation and testing with a ratio of 80:10:10, respectively. We gathered RSS data across two buildings with four floors each, using a smartphone and an app we developed for data collection and real-time inferencing.

3.2. Preprocessing

We evaluate our proposed HyTra networks on the UJI (public) dataset for comparison with existing techniques and a larger private dataset (SPOT) with approximately ten times more training and validation samples. Although we discuss our results from training and testing for both public and private datasets, due to confidentiality, only describe the public dataset in this section. However, the private datasets were processed in a similar manner and differed mainly in the number of samples, unique locations and number of WAPs.

Table 1 summarizes the input and outputs used for fingerprinting. Five hundred twenty (520) WAPs serve as input for each sample, and amongst the nine other identifiers, four are used as labels. Latitude and Longitude values are labels for the regression task and are reported in meters but are not included in Table 1 as classification is the primary objective of this study. Building and floor labels are used for classification and have been assigned a numerical value corresponding to a specific building and floor. Building is set as either 0, 1 or 2, and the floors are originally designated as either 0, 1, 2, 3 or 4.

Table 1. Input and label specification of fingerprinting data.

Input/Label	Fingerprinting Data	Description
Input	$\vec{R} = \{R_0, R_1, \ldots, R_{519}\}$	RSS values from 520 WAPs in dBm.
Classification Label	$B_k, k \in \{0, 1, 2\}$	Building ID, 3 unique buildings.
Classification Label	$F_u, u \in \{0, 1, \ldots, 12\}$	Floor ID, 13 unique floors.

The publicly available UJI dataset does not distinguish between floors across buildings i.e. the label for floor 1 in building 1, 2 and 3 are all the same (the label is 1). The lack of a unique floor mapping can pose as problematic in training a fingerprinting algorithm. We assign a unique label to every floor (shown in Table 1 and Figure 4), ordering the labels by floor first and then buildings; we found this strategy to work best for UJI. The five remaining identifiers are; Relative position, which indicates if the user was situated inside or at the entrance of a given space; UserID, PhoneID and Timestamp, equating to 529 attributes per sample. The remaining identifiers aided during data exploration but were not incorporated when training the HyTra network.

We transform the input and labels into a format that would allow the Transformer network to learn the best representation. Firstly, we replace the out-of-range RSS values (+100 dBm) in the input with -110

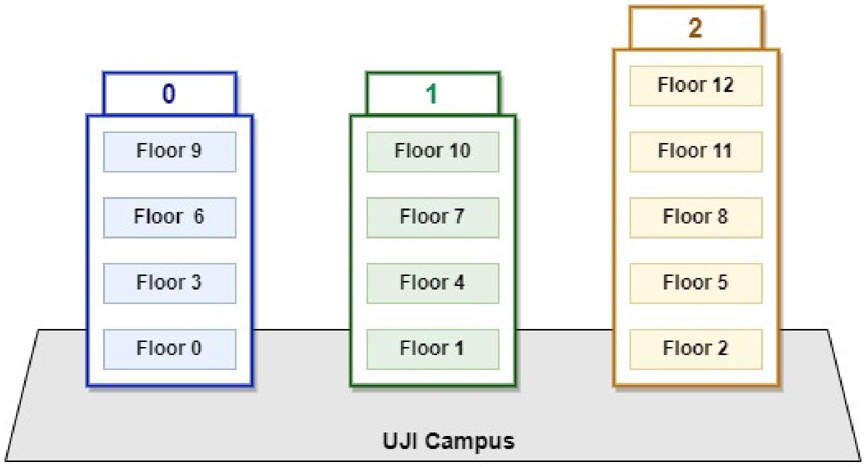


Figure 4. Mapping of building and floor labels to unique floor labels.

dBm, as suggested by [35], since -104 dBm was the weakest signal observed within the training set. Then we apply zero-to-one normalization to obtain the scaled RSS values:

$$\text{Normalized } RSS_{n,i} = \begin{cases} \dfrac{RSS_{n,i} - \min}{-\min}, & \begin{array}{r} 0 \leq i < 520 \\ 0 \leq n < 21,048 \\ -110 \leq RSS_{n,i} \leq 0 \end{array} \end{cases}, \qquad (1)$$

where $RSS_{n,i}$ represents the n^{th} sample (training and test combined) and the RSS value from the i-th WAP, -110 dBm is the min value. Normalization is crucial as it maps features to the same scale, which helps with the trainability of the deep neural network. After normalizing the RSS values, we may apply principal component analysis (PCA) to the input ($\vec{R}$, a one-dimensional array of length 520). As several other methods discussed in related works concluded, the noisy RSS input should be preprocessed to denoise and reduce dimensionality to mitigate overfitting. We test three input transformations to minimize overfitting for the UJI dataset, including PCA.

3.3. HyTra Models

We propose a multi-headed Transformer classification layer (one head per class) to account for the multiple classes (building, floor and room) within a complex indoor environment. We added building, floor, and room-specific learned position embeddings. Figure 5 illustrates our standard HyTra architecture for building-floor-room indoor localization with three different input transformations to optimize our accuracy on the hold-out testing set. Additionally, we proposed a secondary modified HyTra network with multiple stacked Encoders (one for each classification level), vanilla attention (single head) and a FFN as its input transformation, shown in Figure 6. The objective of the modified HyTra network is to explicitly and hierarchically filter the already attention-filtered value (AFV). Thus, we refer to our modified HyTra network as HyTra-HF.

3.3.1. Model Input

After the RSS data has undergone preprocessing, it is sequenced and batched (Figure 7); doing so results in an input shape: *Batch Size* × *Sequence Length* × *Features*. While our definition of Batch Size (number of training samples per iteration) is fixed, *Sequence Length* and *Features* are not. We further elaborate on the two ways we structure our input matrix in Table 2.

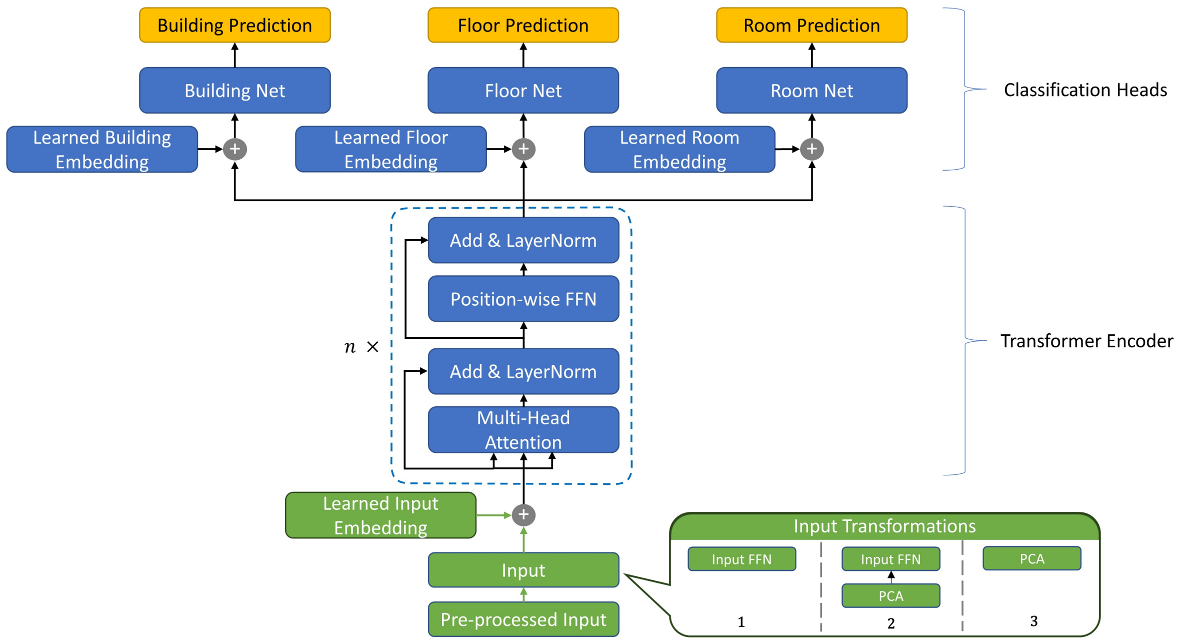


Figure 5. General HyTra network for the indoor localization classification task, consisting of a stacked Encoder with learnable position embeddings and multiple classification heads. Three different input transformations are tested to obtain the best testing set accuracy.

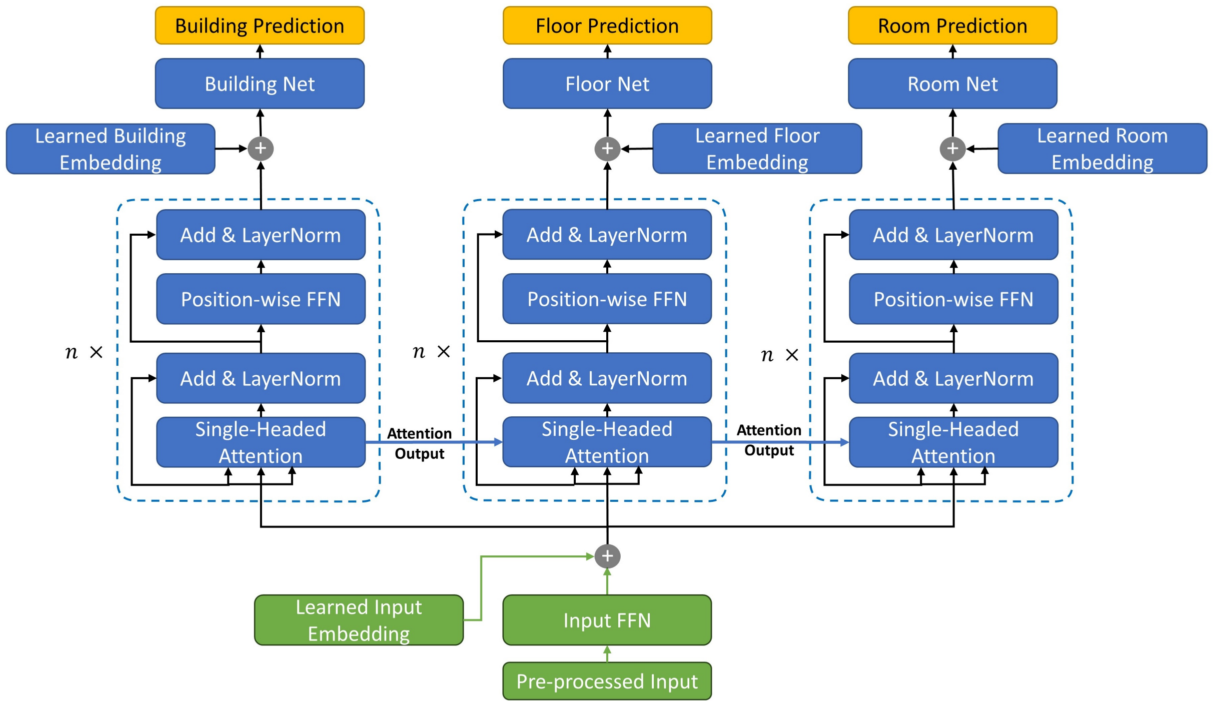


Figure 6. HyTra-HF Network composed of stacked Encoder blocks and a single attention head for each class (building, floor and room). Uses a feed-forward network (FFN) as its input transformation.

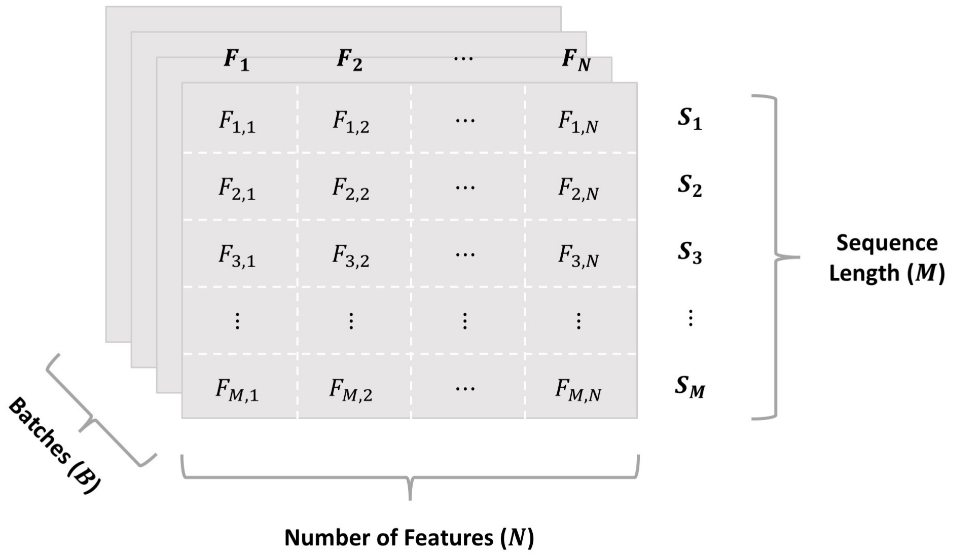

F_1
F_2
$\cdots$
F_N
$F_{1,1}$ $F_{1,2}$ $\cdots$ $F_{1,N}$
$F_{2,1}$ $F_{2,2}$ $\cdots$ $F_{2,N}$
$F_{3,1}$ $F_{3,2}$ $\cdots$ $F_{3,N}$
$F_{M,1}$ $F_{M,2}$ $\cdots$ $F_{M,N}$
S_1
S_2
S_3
S_M
Sequence Length (M)
Batches (B)
Number of Features (N)

Figure 7. Standard batched input matrix representation. Shape: $B \times M \times N$.

3.3.2. Input Transformation

The performance of a Transformer model is highly dependent on the quality of the input embeddings. A rich input embedding provides contextual and relational information or semantic meaning in the NLP case. Instead of individual words, we tokenize the WAP inputs and convert them to d-dimensional embedding vectors. We experiment with a combination of three input transformations and two input methods to obtain the best representation of our input features, as shown in Figure 5 and Table 2, respectively.

Firstly, we experiment with just a feed-forward network (FFN), composed of two linear layers with a ReLU activation in between and a dropout layer afterward. This transformation is primarily intended for input method 1, as there is only a single RSS value per WAP per sample; hence, we must scale up the feature dimension to increase the expressive ability of our network. This transformation is best suited to larger datasets as deep neural networks generally require lots of data to obtain high-fidelity representations.

Our second (2) transformation method applies PCA to the dataset; then, it is structured, batched, and fed through a FFN. The intuition is that PCA reduces the data's dimensionality while retaining the most important components. As discussed in related works, existing techniques utilize methods or networks like the SAE to denoise the data before feeding it to their main classification algorithm.

Lastly, our third (3) input transformation only applies PCA to the data. Since we do not employ a FFN to upscale the feature dimension, we couple this transformation with our second input method, where the PCA-transformed data (principal component scores) serves as the input features. As for our modified network, HyTra-HF, we only employ a FFN network as our input transformation and arrange the RSS data using input method one (1).

3.3.3. Position Embedding

Vaswani et al. [6] originally generated position embeddings via a fixed sinusoidal function, while all our position embeddings are learned through training. We employ learnable position embeddings as they are known to capture positional information and can model implicit complex behaviours, which aids the training process [42]. Thus, we fix the order of WAPs in the input sequence and add the learned positional embeddings before feeding the input to the Transformer and at each classification head (shown in Figures 5

Table 2. Dimension-wise descriptions for both input methods.

Input Method 1	Sequence Length: the total number of WAPs (520 for UJI) or principal components after applying PCA.
	Features: the number of RSS readings per WAP or scores per principal component.
	Objective: to allow the attention mechanism to weigh input based on the similarity of the features and allow the learnable position embedding to learn the relative position of each WAP. This method is preferred for larger datasets, as it would enable the network to compose a good representation of each WAP.
Input Method 2	Sequence Length: number of samples. A single sample can be used (sequence length of 1), but we recommend batching multiple samples chronologically to benefit from the attention mechanism.
	Features: RSS values from each WAP (520 for UJI) or scores of principal components after applying PCA.
	Objective: representing all of the WAPs or principal components as features is preferred for smaller datasets; this would circumvent the need to obtain an accurate input representation which is difficult on smaller datasets (like UJI by deep learning standards).

and 6), which enables the embeddings to capture the order and relative positions of each WAP in a high dimensional space at each level of classification. Adding learned embeddings to the transformed input at each classification head pushes the input vector representations of similar WAPs closer together in the high-dimensional embedding space. Note: applying PCA to reduce dimensions of the RSS input (shortened *sequence length*) maintains the underlying intuition of the learnable embedding; instead of expressing the relative position of WAPs, it represents the similarity of principal components.

3.3.4. Encoder

For the main HyTra network, the Encoder contains two important sublayers, the multi-head attention layer and the pointwise feed-forward network (FFN). We employ scaled dot product attention to measure the similarity between input pairs

$$\text{Attention}\left(Q, K, V\right) = \text{softmax}\left(\frac{Q \cdot K^{\top}}{\sqrt{d_k}}\right) V, \tag{2}$$

where $Q \in \mathbb{R}^{W \times d_k}$ is the query matrix, $K \in \mathbb{R}^{W \times d_k}$ is the key matrix, $V \in \mathbb{R}^{W \times d_v}$ is the value matrix, W is the number of WAPs, d^k and d^v are the dimensionality of the key and value matrix, respectively. The query, key, and value matrices are derived from the input and go through a series of operations to compute the final attention-filtered value matrix (AFV) obtained using equation (2). The AFV matrix indicates to the network which WAPs to give the most importance. Then, the original position-embedded input is re-added to the AFV through a residual connection and normalized. Residual connections help mitigate the vanishing gradient problem by allowing the gradients to flow through the network without passing through non-linear activation functions. They also reinject positional information, which may fade as the input flows through complex layers. Multiple attention heads (single head refers to single-scaled dot-product operation) combine multiple attention mechanisms in parallel by concatenation and then are linearly transformed to match the original size.

The AFV is then fed to the pointwise feed-forward network [6], composed of two linear transformations with a ReLU activation in between and is applied to each position separately and identically. For the same reasons stated above, the AFV is added to the output of the FFN via residual connection and normalized before heading to the classification heads. The Encoder is stacked n times with identical layers, where n is a hyperparameter, allowing the Encoder to model complex behaviour and extract hierarchical features.

3.3.5. Encoder for HyTra-HF

We propose storing the attention outputs and passing them to the subsequent Encoder of the succeeding class, passed from building to floor to room. To implement this, we need our attention heads to match; however, for Multi-Headed-Attention (MHA), each attention head may focus on different parts of the input. To avoid this mismatch, we rely on vanilla attention (a single attention head) supported by the observations of Paul et al. [43], indicating a majority of attention heads can be removed without compromising accuracy.

$$\text{Attention}\left(Q, K, V, A_{\text{prev}}\right) = A_{\text{prev}} \cdot \text{softmax}\left(\frac{Q \cdot K^{\top}}{\sqrt{d_k}}\right) V, \tag{3}$$

where $A_{\text{prev}} \in \mathbb{R}^{W \times W}$ is the attention output from the previous Encoder, W is the number of WAPs, and the resulting AFV is of shape $W \times d_v$, identical to standard HyTra.

3.3.6. Classification Heads

The Encoder obtains the most important features from the input, and then each classification head learns a mapping from these extracted features to their respective class. We add an embedding layer to each classification head to learn the relative position of each WAP at a building, floor and room level. Each classification head has a pointwise feed-forward network to aid the network in learning the complex mapping of AFV to building, floor and room. Lastly, a one-by-one convolution is applied over the input sequence length, extracting salient features across all WAPs and reducing the length to one. The resultant class of a given input is the index of the largest value in the output matrix, which is obtained using Argmax.

3.4. Training Process

The training process is briefly outlined by Algorithm 1 and the hyperparameters for the HyTra network are listed in Table 3. We manually tune the values of the stated hyperparameters to settle on a configuration that delivers the best-performing solution. To accelerate the gradient descent algorithm, we use the AdamW [44] optimizer with a Cosine learning rate scheduler (with warmup), which combines both warmup and learning rate decay.

3.5. System Overview

Figure 8 presents a system overview of the indoor localization algorithm using the proposed HyTra model for the classification task. Similar to other WiFi fingerprinting methods, our approach comprises of two phases: offline and online phase. Both phases require preprocessing the RSS data by replacing out-of-range values and applying zero-to-one normalization to bring RSS values to the same scale. During the offline phase, the HyTra model is trained using the preprocessed RSS data; the provided building labels and unique floor labels are used for classification. RSS data collected during the online phase undergoes the same preprocessing steps to obtain building and floor predictions. We trained and tested the proposed networks on Google Colab, a cloud-based Jupyter notebook environment with limited free GPU or unlimited paid access. Our setup utilized a Tesla T4 GPU, Intel (R) Xeon (R) CPU, Python-3.8.10 and Pytorch Lightning 1.9.3 for developing, training, and testing our proposed networks. The trained HyTra model predicts the user's location using RSS data collected and preprocessed during the online phase.

Algorithm 1: Pseudo code to train the HyTra network on UJI

```
Input: $\vec{R}$
Output: Predicted Location of User $(B_{predict}, F_{predict})$
 1: if $(B_k, F_l)$ are the given labels then
 2:     Compute unique floor labels $(F_u)$
 3: end if
 4: for RSS in $\vec{R}$ do
 5:     if $RSS == 100$ then
 6:         Replace $RSS$ with $-110$
 7:     end if
 8: end for
 9: $\hat{R} = ZeroToOneNormalization(\vec{R})$
10: $Train, Test, Validation = Split(\hat{R}, stratify = F_u)$
11: Train HyTra
12: Finetune HyTra
13: return $(B_{predict}, F_{predict})$
```

Table 3. Hyperparameters used to train HyTra and HyTra-HF.

Parameter	Value	Description
Learning Rate	0.0003	Step size of parameter update.
Scheduler	Cosine with Warmup	Adjusts learning rate during training.
Encoder Layers	1 & 2	Number of stacked identical encoder layers.
Attention Heads	8	Number of query, key and value pairs used for MHA.
Model Dimensions	256	Dimensionality of the transformed input features.
Dropout Rate	0.2	Fraction of inputs dropped during training.
Batch Size	128	Number of samples per iteration.
Optimizer	AdamW	Stochastic gradient descent method to update model parameters.
Loss Function	Cross-Entropy	Criterion used to evaluate classification performance.
PCA Components	128	The number of components used to represent directions of maximum variance in the data.

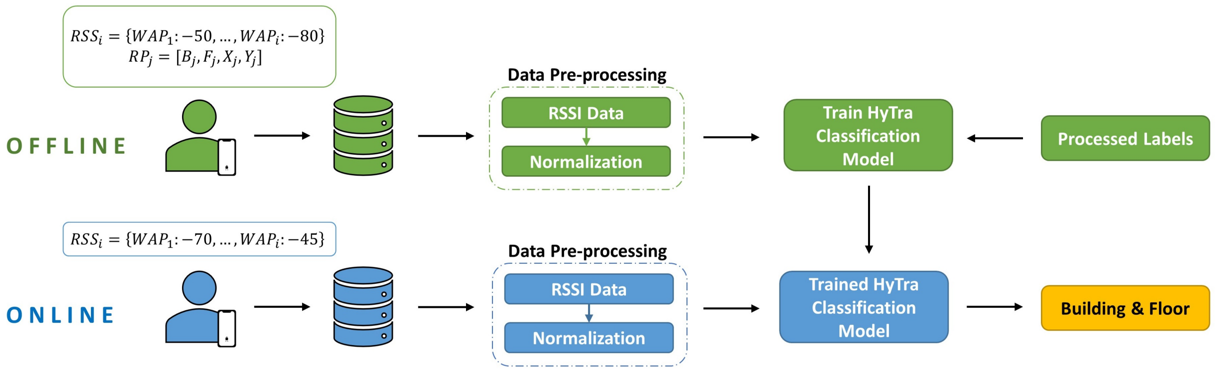

$RSS_i = \{WAP_1: -50, \ldots, WAP_i: -80\}$
$RP_j = [B_j, F_j, X_j, Y_j]$
OFFLINE
Data Pre-processing
RSSI Data
Normalization
Train HyTra Classification Model
Processed Labels
$RSS_i = \{WAP_1: -70, \ldots, WAP_i: -45\}$
ONLINE
Data Pre-processing
RSSI Data
Normalization
Trained HyTra Classification Model
Building & Floor

Figure 8. Overview of indoor localization scheme utilizing the proposed HyTra model for classification.

4. Results and Discussion

In this section, we evaluate and compare the classification performance of the proposed networks across datasets by testing on public (UJI) and private datasets (SPOT and UTP). Then, to determine the effectiveness of hierarchical filtering, we analyse the confusion matrix of HyTra and HyTra-HF. Lastly, we compare our proposed networks against established deep learning-based techniques identified in the related works section on the UJI dataset.

4.1. Performance comparison of ML and DL methods on SPOT

Since SPOT is our largest but private dataset, we tested a popular ML and DL network each on the SPOT dataset to serve as baselines for comparison against our proposed networks. We selected Random Forests [45] which utilize a fundamentally different computational element, decision trees,in contrast with perceptrons used in neural networks. We also chose ResNet-34 [46] with 1-D convolutions to investigate any advantages of local feature aggregation over scale-free feature aggregation in transformer networks. ResNets acquire features from adjacent RSS readings in the input sequence at the initial layers of the network and learn global features at the downstream layers. In contrast, transformer networks utilize a key, query and value framework which extracts features from RSS word pairs across all spatial distance scales.

As Table 4 indicates, transformer networks outperform convolutional networks. HyTra and HyTra-HF marginally outperforms a conventional transformer network. Random Forests offers competitive accuracy at the cost of model size which is two orders of magnitude larger (see 9).

Table 4. Classification results for different ML and DL methods on the SPOT dataset.

Model	Classification Accuracy		
	Building Accuracy	Floor Accuracy	Room Accuracy
Random Forests (100 Trees)	99.98%	99.19%	97.46%
ResNet34	-	-	94.94%
Standard Transformer (Encoder)	-	-	97.60%
HyTra	100.0%	99.41%	97.87%
HyTra-HF	99.99%	99.40%	97.79%

4.2. HyTra Comparison on Public and Private Datasets

Here we compare the classification performance of the HyTra network across three datasets, as indicated by Table 5. The UJI dataset does not contain room-level labels in the testing set but our private datasets do. HyTra obtains nearly 100% building accuracy on all three datasets and achieved notably improved floor accurate on the largest SPOT dataset. A larger dataset allows HyTra to learn a more accurate representation of each WAP, improving its performance as a result. The size of the training set and the data collection strategy for both training and testing sets plays a significant role in model performance across datasets.

Table 5. HyTra classification results using the UJI and two private (SPOT and UTP) testing datasets.

Datasets	Training Samples	Classification Accuracy		
		Building Accuracy	Floor Accuracy	Room Accuracy
SPOT	165 K	100%	99.41%	97.87%
UJI	17.9 K	99.82%	95.14%	-
UTP	0.832 K	100%	97.10%	78.60%

Comparing the classification accuracy for each data split on the UJI dataset (Table 6), we see the HyTra models overfits on the training set. There is a significant difference between training and testing floor accuracy for all three input variants. PCA with FFN (Input Transform 2) performs the worst out of the three input transformation methods, probably because the FFN is restricted from learning a general representation based on the lower-rank estimate from PCA. Meanwhile, just applying PCA (input transform three) obtains a slightly better testing set accuracy on the UJI dataset when paired with the second input method.

We believe this is due to a combination of reduction in noise from applying PCA and treating the PCA-transformed features directly as the input embedding (Input Method 2), circumventing HyTra from learning a feature embedding vector. However, the best accuracy is obtained when we utilse input method and transform one (1,1), indicating a better input representation is obtained by feeding the network all 520 WAPs and allowing it to extract the features it finds the most essential. Interestingly, we notice a similarly large gap in performance on the testing set among others using UJI to test their deep learning-based methods [29, 35]. Moreover, like [29], we had to apply an aggressive dropout rate of 0.7 and used a single Encoder layer to reduce overfitting on the UJI training set. We hypothesize one of those reasons is that roughly 11% (55 out of 520) of the WAPs in the training set are null, as they were only observed during testing, which results in the network ignoring those 55 WAPs during training; thus, failing to learn a useful embedding vector for the missing WAPs.

4.3. Comparison of HyTra-HF on Public and Private Datasets

Here we compare the classification performance of the HyTra-HF network on three different datasets, highlighted by Table 7. We see the modified HyTra method obtains similar results to our standard HyTra method on SPOT, as mentioned previously. Moreover, the results for UJI show a significant improvement in floor accuracy (96.7%) compared with HyTra, which achieves 94.3% for floor accuracy, demonstrating the effectiveness of HyTra-HF in filtering out troublesome WAPs. The overall results for HyTra-HF indicate that hierarchically filtering the AFV, the preceding class itself must be accurate, or it may negatively impact the filtered class, as in the case of the UTP dataset. Like the standard HyTra, HyTra-HF achieves better floor accuracy on the UJI dataset without using a denoising method such as PCA, using both first input method and input transform.

Table 6. Classification results for each data split on the UJI dataset for HyTra with the different input methods and transformations. HyTra (Input Method, Input Transformation), ordering as specified in the methodology section.

Model	Dataset Split	Classification Accuracy	
		Building Accuracy	Floor Accuracy
HyTra (1,1)	Training	100%	99.12%
	Validation	99.80%	99.65%
	Testing	99.82%	95.14%
HyTra (1,2)	Training	99.92%	99.08%
	Validation	99.85%	99.10%
	Testing	99.91%	88.12%
HyTra (2,2)	Training	99.87%	98.45%
	Validation	99.85%	98.95%
	Testing	99.82%	93.44%
HyTra (2,3)	Training	99.79%	99.09%
	Validation	99.75%	99.20%
	Testing	99.91%	94.33%

4.4. Comparison of Confusion Matrix

To understand the impact of hierarchical filtering, we compare the confusion matrix of HyTra-HF with HyTra (1,1) at each classification level (building, floor, and room) since both methods utilise the same input method and transformation. The confusion matrix presents a visual summary of the classification performance of a network on the testing set of the respective dataset.

4.4.1. SPOT

First, we evaluate using the SPOT dataset. We notice no improvement in building (Figure 9) and floor classification (Figure 10) but do so for room classification; this could result from challenging signal conditions leading to a higher percentage of mispredictions occurring in building three (3). Furthermore, we notice for HyTra-HF, the single mispredicted class for room classification is also mispredicted at building and floor classification; however, if there is no overlapping misprediction for building and floor, HyTra-HF predicts the room in the correct building, as seen in Figure 11.

Table 7. HyTra-HF classification results using the UJI and two private (SPOT and UTP) testing datasets.

Datasets	Training Samples	Classification Accuracy		
		Building Accuracy	Floor Accuracy	Room Accuracy
SPOT	165 K	100%	99.4%	97.8%
UJI	17.9 K	99.7%	96.7%	-
UTP	0.832 K	100%	97.1%	76.7%

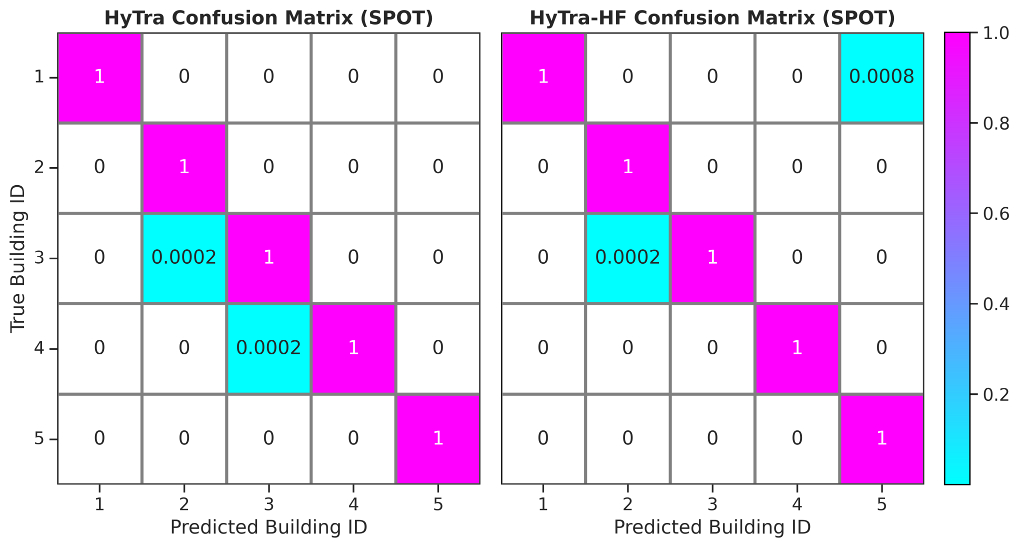


Figure 9. Comparison of confusion matrix between HyTra (Left) and HyTra-HF (Right) for Building-Level classification on the SPOT Dataset.

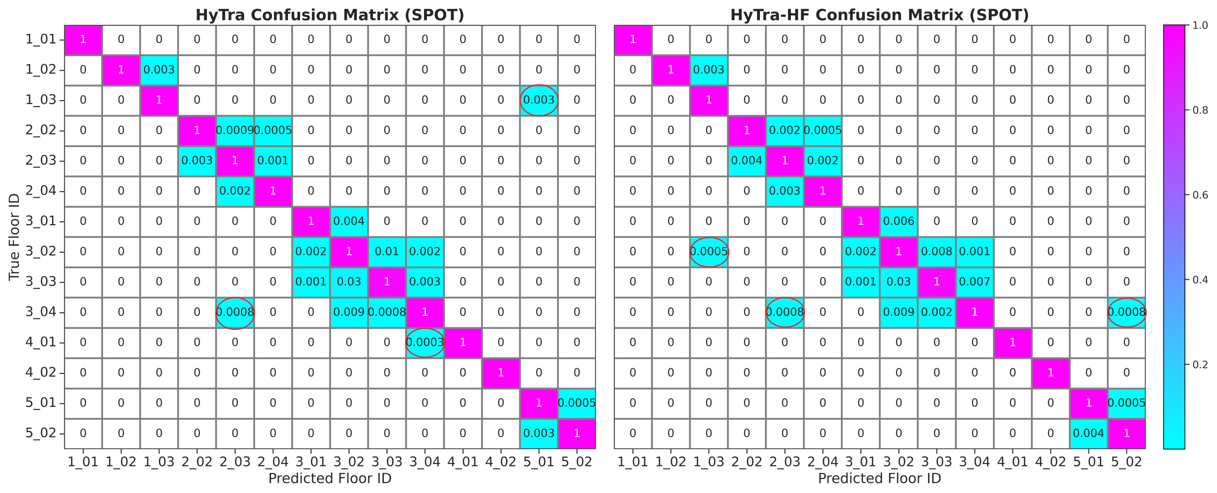


Figure 10. Comparison of confusion matrix between HyTra (Left) and HyTra-HF (Right) for Floor-Level classification on the SPOT Dataset.

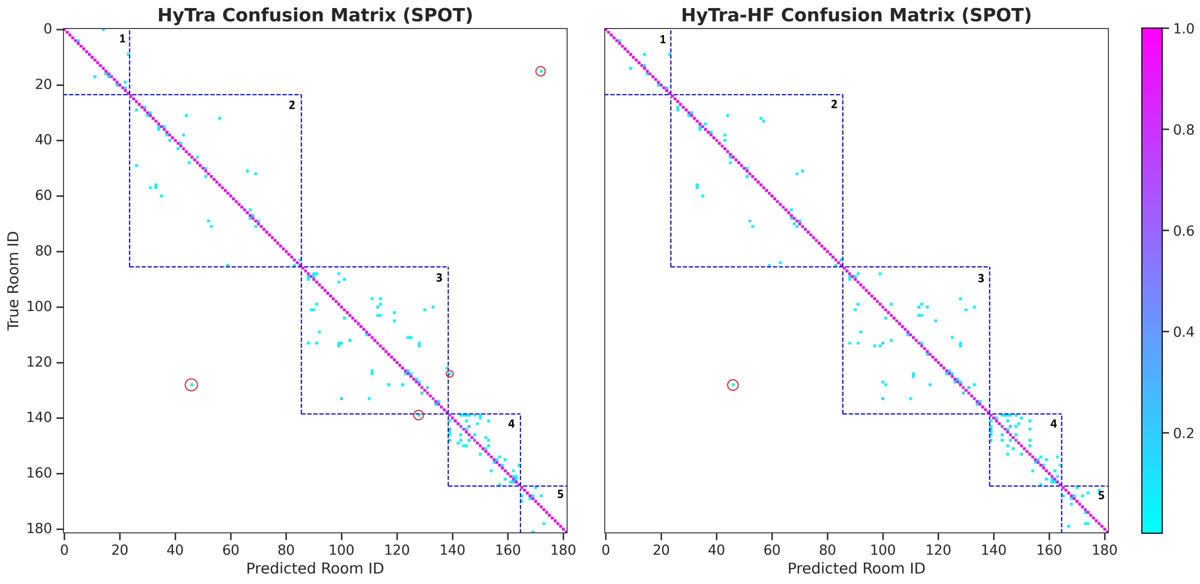


Figure 11. Comparison of confusion matrix between HyTra (Left) and HyTra-HF (Right) for Room-Level classification on the SPOT Dataset. Room IDs are arranged and segmented by Building ID in ascending order. Misclassifications across buildings are circled in red.

4.4.2. UJIIndoorLoc (UJI)

The results for UJI highlight, regardless of a misprediction for building one (1) as shown in Figure 12, HyTra-HF can effectively filter troublesome WAPs to reduce and completely avoid making mispredictions across buildings, as observed when comparing the floor confusion matrices in Figure 13. Filtering WAPs is particularly effective for UJI, given 11% (55 out of 520) of WAPs are not present during training, and only 60% (312 out of 520) of WAPs are shared between training and testing (validation originally).

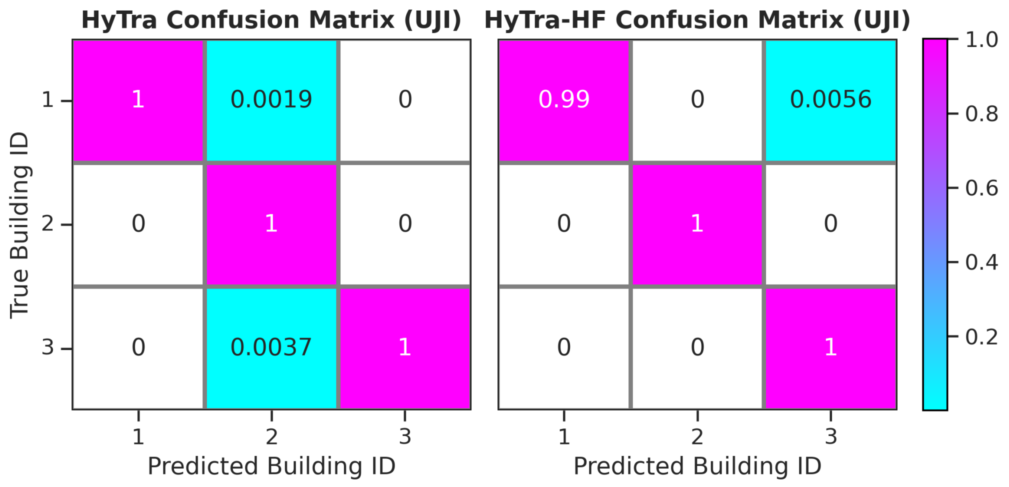


Figure 12. Comparison of confusion matrix between HyTra (Left) and HyTra-HF (Right) for Building-Level classification on the UJI Dataset.

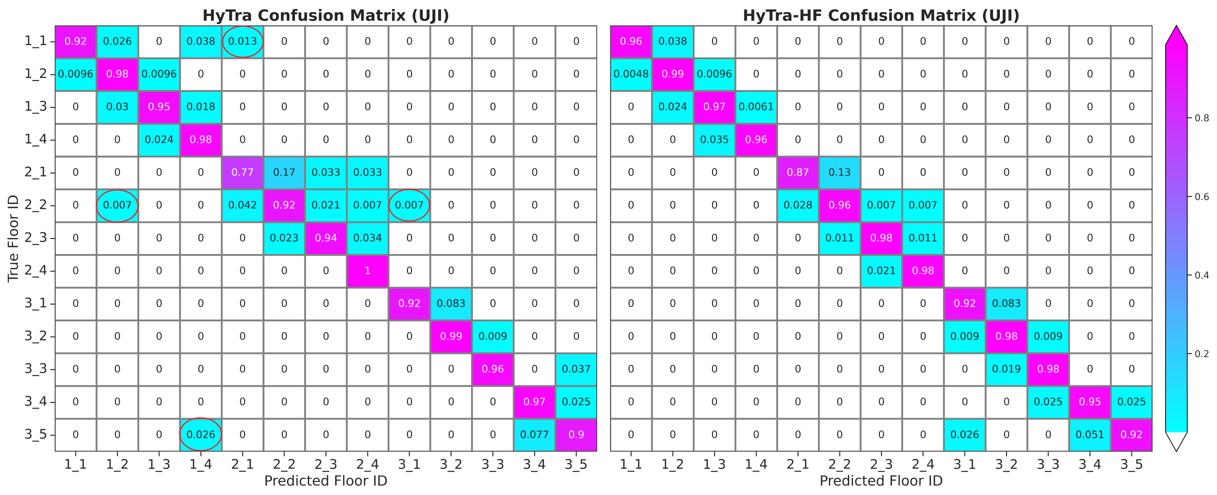


Figure 13. Comparison of confusion matrix between HyTra (Left) and HyTra-HF (Right) for Floor-Level classification on the UJI Dataset. Misclassifications across buildings are circled in red.

4.5. Comparison of Classification Results

In this section, we compare the classification accuracy of the proposed networks with deep learning-based solutions from literature on the UJI dataset, shown in Table 8. Existing methods listed in Table 8 employ an autoencoder-based network to reduce the input dimension and extract essential features; we opted for PCA for our standard HyTra network to prevent overfitting our reduced input representation on the UJI training set, which can occur for gradient-based methods like the autoencoder. On the contrary, HyTra-HF attains the highest floor accuracy without needing to denoise the input. Our best results were obtained using just the FFN, allowing the network to extract meaningful features. Compared to all the listed methods, HyTra-HF achieves the highest floor accuracy (96.67%) by utilizing a simple FFN to produce feature embeddings, which are refined through the training process and filtered hierarchically via attention outputs. For reference, the standard Transformer with just an Encoder, without the proposed learnable embeddings, multiple classification heads, or hierarchical filtering obtains 93.16% for floor accuracy, a significant reduction from our proposed networks.

Table 8. Comparison of classification results of existing methods HyTra-HF on the UJI dataset.

Deep Learning Methods	Classification Accuracy	
	Building Accuracy	Floor Accuracy
DNN [35]	-	91.10%
Scalable DNN [36]	99.50%	91.26%
Standard Transformer (Encoder)	-	93.16%
Hierarchical RNN [41]	100%	95.23%
2D-CNN (m-CEL) [40]	-	95.30%
CNNLoc [29]	100%	96.03%
HyTra-HF	99.71%	96.67%

4.6. Comparison of Model Size and Computational Cost

We compare the trainable parameters (Table 9) of the proposed methods and several HyTra-HF variants against methods from the literature trained on UJI, whose parameters were obtained from GitHub or inferred from their model architecture stated in their respective article. For neural networks, the trainable parameters are weights and biases of the perceptrons. The number of trainable parameters is representative of model size but only indicative of the computational complexity because each unique trainable parameter is associated with at least one multiplication or addition operation. Convolutional networks employ repeated weights and perform orders-of-magnitude more computations (multiplications and additions) than the number of trainable parameters.

Our standard HyTra-HF contains almost an order of magnitude more trainable parameters than the best performing method reported in literature. The larger model size could possibly account for the difference in performance so, we conducted further experiments with the model dimension and the number of Encoder layers, combined with a sparse weight initialization scheme [47], to obtain three (3) HyTra-HF variants at comparable parameters to existing methods.

HyTra-HF, with a model dimension (dim) of 64, outperforms most existing methods in floor accuracy at fewer parameters but falls short compared with CNNLoc [29]. Additionally, we employ a sparse weight initialization scheme, sparsity-aware orthogonalization (SAO [47]), to reduce even more weights while assuring connectivity and approximate dynamical isometry. Orthogonalizing the weights pushes the singular values of the Jacobian to be close to one (1), which aids the learning dynamics of the network; this property is referred to as dynamical isometry [48]. We apply SAO to HyTra-HF (64-dim) and experiment

Table 9. Comparison of trainable parameters and model size between existing and proposed methods. Asterisk (*) denotes HyTra-HF pruned using SAO [47] with d indicating the number of connections, L is the number of Encoder layers, and dim is the model dimension.

Model	Test Accuracy		Trainable Parameters	Model Size (MB)
	Building	Floor		
DNN [35]	-	91.10%	397 K	-
Scalable DNN [36]	99.50%	91.26%	395 K	-
Hierarchical RNN [41]	100%	95.23%	409 K	-
2D-CNN (m-CEL) [40]	-	95.30%	1.11 M	-
CNNLoc [29]	100%	96.03%	346 K	-
Random Forests	99.91%	90.73%	1000 Trees	255.7
Standard Transformer (Encoder)	-	93.16%	284 K	1.1
ResNet34	-	92.17%	6.35 M	24.9
HyTra (1, 1)	99.82%	95.14%	1.19 M	6.1
HyTra-HF	99.73%	96.67%	2.42 M	9.2
HyTra-HF (64-dim, 1L)	99.73%	95.41%	261 K	1.1
*HyTra-HF (64-dim, 4L, 8-d)	99.91%	95.05%	198 K	2.3
*HyTra-HF (64-dim, 1L, 32-d)	99.55%	95.95%	198 K	1.1

with single (1L) and four (4L) Encoder layers at different sparsity settings to achieve a low parameter count. Small parameter count and model size is advantageous for deployments on edge-computing devices.

5. Conclusion

Location categorization can be framed as a hierarchical classification task where each class is a progressively precise position in space i.e. building, floor and room. In this work we attempt to exploit this hierarchical relationship between classes to improve classification accuracy in complex indoor environments using RSS WiFi fingerprinting. We propose a Transformer-based Encoder-Only neural network with multiple classification heads for progressively refined location categorization, each with their own learnable embedding, over and above the learnable embedding of the RSS signal input. We investigated 2 variants which are the HyTra network comprising a shared feature representation layer feeding into the multiple classification heads and HyTra-HF with each classification head having its own feature representation and learnable embedding. In Hytra-HF, hierarchical coupling between coarse and fine location features are achieved by applying attention output from the coarse features to the attention-filtered values (AFV) of fine features. We evaluated the performance of both variants on 3 datasets with approximate order of magnitude differences in size to study the performance of our method with dataset scaling. HyTra consistently outperforms alternative machine learning (Random Forest) and deep learning approaches (convolutional ResNet34) as well as other deep neural network methods reported from literature and its performance improves with larger training dataset size. The performance of HyTra is sensitive to the quality of the high-dimensional input embedding (FFN) stage and care must be taken to train or optimize this stage to enhance the benefits of the hierarchical learnable embeddings. When applied in conjunction with Sparsity Aware Orthogonal (SAO) initialization, HyTra achieves impressive model compression by almost an order of magnitude without significant reduction of accuracy and attains best-in-class accuracy at almost half the state-of-art model size. Our transformer architecture is potentially generalizable to other multi-class classification tasks where the classes enjoy a hierarchical rather than exclusive relationship such as taxonomies of language, plants and animals, knowledge or disease. Future work will include investigations to validate if the HyTra architecture generalizes to other types of hierarchical classifications and to enhance the separability of common and specialized features within the hierarchical learnable embeddings.

Acknowledgments

Funding: This research was funded by the Petroleum Research Fund of Malaysia, grant number E.025.FOF.02021.0.14 awarded to PETRONAS Research Sdn Bhd and in collaboration with Universiti Teknologi PETRONAS.

Author contributions: Conceptualization: Muneeb Nasir, Mazlaini Yahya, Afidalina Tumian and Eric Tatt Wei Ho; Data curation: Mazlaini Yahya; Formal analysis: Muneeb Nasir and Eric Tatt Wei Ho; Funding acquisition: Tong Boon Tang, Mazlaini Yahya, Afidalina Tumian and Eric Tatt Wei Ho; Investigation: Muneeb Nasir, Ibrahima Faye and Afidalina Tumian; Methodology: Muneeb Nasir, Kiara Esguerra, Ibrahima Faye, Mazlaini Yahya, Afidalina Tumian and Eric Tatt Wei Ho; Project administration: Tong Boon Tang, Afidalina Tumian and Eric Tatt Wei Ho; Resources: Tong Boon Tang, Mazlaini Yahya and Afidalina Tumian; Software: Muneeb Nasir and Kiara Esguerra; Supervision: Ibrahima Faye, Tong Boon Tang, Afidalina Tumian and Eric Tatt Wei Ho; Validation: Muneeb Nasir and Kiara Esguerra; Visualization: Muneeb Nasir; Writing – original draft: Muneeb Nasir; Writing – review and editing: Mazlaini Yahya, Afidalina Tumian and Eric Tatt Wei Ho.

Disclosure statement: The authors declare no conflict of interest.

References

[1] Germán Martín Mendoza-Silva, Joaquín Torres-Sospedra, and Joaquín Huerta. A meta-review of indoor positioning systems. *Sensors*, 19(20), 2019.

[2] George Sithole and Sisi Zlatanova. Position, location, place and area: An indoor perspective. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, III-4:89–96, 2016.

[3] Location-based services (lbs) market size and forecast - 2030, 2023. https://www.alliedmarketresearch.com/location-based-services-market, Accessed on 20.09.23.

[4] Shuang Shang and Lixing Wang. Overview of wifi fingerprinting-based indoor positioning. *IET Communications*, 16:725–733, 4 2022.

[5] Oludare Isaac Abiodun, Aman Jantan, Abiodun Esther Omolara, Kemi Victoria Dada, Abubakar Malah Umar, Okafor Uchenwa Linus, Humaira Arshad, Abdullahi Aminu Kazaure, Usman Gana, and Muhammad Ubale Kiru. Comprehensive review of artificial neural network applications to pattern recognition. *IEEE access*, 7:158820–158846, 2019.

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010. Curran Associates Inc., 2017.

[7] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics.

[8] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997.

[9] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[10] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, volume 1, pages 4171–4186, 2019.

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[13] Zhongfeng Zhang, Hongxin Du, Seungwon Choi, and Sung Ho Cho. Tips: Transformer based indoor positioning system using both csi and doa of wifi signal. *IEEE Access*, 10:111363–111376, 2022.

[14] Wen Liu, Qianqian Cheng, Zhongliang Deng, Hong Chen, Xiao Fu, Xinyu Zheng, Shixuan Zheng, Cunzhe Chen, and Shuo Wang. Survey on csi-based indoor positioning systems and recent advances. In *2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8, 2019.

[15] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650. Association for Computational Linguistics, 2019.

[16] Yann LeCun, John S. Denker, and Sara A. Solla. Optimal brain damage. In *Proceedings of the 2nd International Conference on Neural Information Processing Systems*, pages 598–605, 1989.

[17] Babak Hassibi, David G Stork, and Gregory J Wolff. Optimal brain surgeon and general network pruning. In *IEEE International Conference on Neural Networks*, volume 1, pages 293–299, 1993.

[18] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. pages 1–42, 2019.

[19] Hidenori Tanaka, Daniel Kunin, Daniel L K Yamins, and Surya Ganguli. Pruning neural networks without any data by iteratively conserving synaptic flow. 2020.

[20] Wei Zhang, Lu Hou, Yichun Yin, Lifeng Shang, Xiao Chen, Xin Jiang, and Qun Liu. TernaryBERT: Distillation-aware ultra-low bit BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 509–521. Association for Computational Linguistics, 2020.

[21] Faheem Zafari, Athanasios Gkelias, and Kin K. Leung. A survey of indoor localization systems and technologies. *IEEE Communications Surveys & Tutorials*, 21(3):2568–2599, 2019.

[22] Navneet Singh, Sangho Choe, and Rajiv Punmiya. Machine learning based indoor localization using wi-fi rssi fingerprints: An overview. *IEEE Access*, 9:127150–127174, 2021.

[23] X. Feng, K. A. Nguyen, and Z. Luo. A survey of deep learning approaches for wifi-based indoor positioning. *Journal of Information and Telecommunication*, 6:163–216, 2022.

[24] Joaquín Torres-Sospedra, Raúl Montoliu, Adolfo Martínez-Usó, Joan P. Avariento, Tomás J. Arnau, Mauri Benedito-Bordonau, and Joaquín 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)*, pages 261–270, 2014.

[25] Yasmine Rezgui, Ling Pei, Xin Chen, Fei Wen, and Chen Han. An efficient normalized rank based svm for room level indoor wifi localization with diverse devices. *Mobile Information Systems*, 2017:1–19, 2017.

[26] Navneet Singh, Sangho Choe, Rajiv Punmiya, and Navneesh Kaur. Xgbloc: Xgboost-based indoor localization in multi-building multi-floor environments. *Sensors*, 22:1–17, 2022.

[27] Roberto Battiti, Nhat Thang Le, and Alessandro Villani. Location-aware computing: A neural network model for determining location in wireless lans. 2002.

[28] Zhenyu Liu, Bin Dai, Xiang Wan, and Xueyi Li. Hybrid wireless fingerprint indoor localization method based on a convolutional neural network. *Sensors*, 19(20), 2019.

[29] Xudong Song, Xiaochen Fan, Chaocan Xiang, Qianwen Ye, Leyu Liu, Zumin Wang, Xiangjian He, Ning Yang, and Gengfa Fang. A novel convolutional neural network based indoor localization framework with wifi fingerprinting. *IEEE Access*, 7:110698–110709, 2019.

[30] Yuan Lukito and Antonius Rachmat Chrismanto. Recurrent neural networks model for wifi-based indoor positioning system. In *2017 International Conference on Smart Cities, Automation & Intelligent Computing Systems (ICON-SONICS)*, pages 121–125, 2017.

[31] Minh Tu Hoang, Brosnan Yuen, Xiaodai Dong, Tao Lu, Robert Westendorp, and Kishore Reddy. Recurrent neural networks for accurate rssi indoor localization. *IEEE Internet of Things Journal*, 6(6):10639–10651, 2019.

[32] Zhenghua Chen, Han Zou, Jian Fei Yang, Hao Jiang, and Lihua Xie. Wifi fingerprinting indoor localization using local feature-based deep lstm. *IEEE Systems Journal*, 14:3001–3010, 6 2020.

[33] Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[34] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(110):3371–3408, 2010.

[35] Michał Nowicki and Jan Wietrzykowski. Low-effort place recognition with wifi fingerprints using deep learning. In *Automation 2017*, pages 575–584. Springer International Publishing, 2017.

[36] Kyeong Soo Kim, Sanghyuk Lee, and Kaizhu Huang. A scalable deep neural network architecture for multi-building and multi-floor indoor localization based on wi-fi fingerprinting. *Big Data Analytics*, 3(1):1–17, 2018.

[37] Feng Qin, Tao Zuo, and Xing Wang. Ccpos: Wifi fingerprint indoor positioning system based on cdae-cnn. *Sensors*, 21(4):1–17, 2021.

[38] Raul Montoliu, E Sansano, Joaquin Torres-Sospedra, and Oscar Belmonte. Indoorloc platform: A public repository for comparing and evaluating indoor positioning systems. In *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8. IEEE, 11 2017.

[39] Marius Laska and Jörg Blankenbach. Deeplocbox: Reliable fingerprinting-based indoor area localization. *Sensors*, 21(6):1–23, 2021.

[40] Marius Laska and Jorg Blankenbach. Multi-task neural network for position estimation in large-scale indoor environments. *IEEE Access*, 10:26024–26032, 2022.

[41] Abdalla Elmokhtar Ahmed Elesawi and Kyeong Soo 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)*, pages 193–196. IEEE Computer Society, 2021.

[42] Yu-An Wang and Yun-Nung Chen. What do position embeddings learn? an empirical study of pre-trained language model positional encoding. *arXiv preprint arXiv:2010.04903*, 2020.

[43] Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[44] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[45] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[46] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[47] Kiara Esguerra, Muneeb Nasir, Tong Boon Tang, Afidalina Tumian, and Eric Tatt Wei Ho. Sparsity-aware orthogonal initialization of deep neural networks. *IEEE Access*, 11:74165–74181, 2023.

[48] Jeffrey Pennington, Samuel S. Schoenholz, and Surya Ganguli. Resurrecting the sigmoid in deep learning through dynamical isometry: Theory and practice. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4788–4798. Curran Associates Inc., 2017.