Human Sensing Indoors in RF

Utilising Unlabeled Sensor Streams

By

JONAS PAULAVIČIUS

Department of Electrical and Electronic Engineering
UNIVERSITY OF BRISTOL

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of MASTER OF SCIENCE BY RESEARCH in the Faculty of Engineering.

APRIL 2023

Word count: 16,729

Abstract

Indoor human sensing in radio frequencies is crucial for non-invasive, privacy-preserving digital healthcare, and machine learning is the backbone of such systems. Changes in the environment affect negatively the quality of learned mappings, which necessitates a semi-supervised approach that makes use of the unlabeled data stream to allow the learner to refine their hypothesis with time.

We first explore the ambulation classification problem with frequency modulated continuous wave (FMCW) radar, replacing manual feature engineering by inductive bias in architectural choices of the neural network. We demonstrate that key ambulations: walk, bend, sit to stand and stand to sit can be distinguished with high accuracy. We then apply variational autoencoders to explore unsupervised localisation in synthetic grayscale images, finding that the goal is achievable with the choice of encoder that encodes temporal structure.

Next, we evaluate temporal contrastive learning as the method of using unlabeled sensor streams in fingerprinting localisation, finding that it is a reliable method of defining a notion of pairwise distance on the data in that it improves the classification using the nearest neighbour classifier by both reducing the number of other-class items in same-class clusters, and increasing the pairwise distance contrast. Compared to the state of the art in fingerprinting localisation indoors, our contribution is that we successfully address the unsupervised domain adaptation problem.

Finally, we raise the hypothesis that some knowledge can be shared between learners in different houses in a privacy-preserving manner. We adapt federated learning (FL) to the multi-residence indoor localisation scenario, which has not been done before, and propose a local fine-tuning algorithm with acceptance based on local validation error improvement. We find the tuned FL each client has a better personalised model compared to benchmark FL while keeping learning dynamics smooth for all clients.

Dedication and acknowledgements

I thank my supervisor, Robert Piechocki, for giving me this opportunity to carry out research independently, it has given me a lot - most importantly I learned the necessity of collaboration and regular two-way feedback with someone researching the same field. I also thank my second supervisor, Seifallah Jardak, who helped me get through putting out my first publication.

My gratitude also extends to Ryan McConville, who was always there with timely help and advice, and to Junaid Bocus for his hands-on approach and always taking time to write some code for me. Much appreciated are Raul Santos-Rodriguez, who was always a positive voice in our irregular meetings, and Jonathan Thomas - for our irregular virtual coffees and for organising RL meetings.

Finally, I thank Toshiba Research Europe Limited for funding me under the EPSRC iCASE scholarship.

AUTHOR'S DECLARATION

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNED: ..... DATE: .....

Table of Contents

List of Tables

List of Figures

Figure **Page**

CHAPTER 1

INTRODUCTION

As the human population ages, it becomes more difficult to provide personal care at the institutional level [1]. Indoor sensing can provide location and activity information which can be used to ease healthcare burden or gain valuable insight into living with various conditions [2, 3]. Promising applications for indoor localisation and activity recognition include the early diagnosis of cognitive disorders such as dementia [4], as well as furthering our understanding of living with Alzheimer's. Indoor localisation based on pervasive radio frequency (RF) technologies is likely to be a key component of digital home-based healthcare, as well as in other health-focused applications such as assisting people with visual impairments inside buildings.

Sensing at radio frequencies has several benefits, with privacy one of the main ones, as light is not required and only a coarse "image" may be formed [5]. In localisation, the coarseness of the labels if WiFi is used is typically room-level location (see [6] for a recent review), though potentially information such as presence can be inferred with access to the signal only [7], without labeled data. Another benefit is that the signal can permeate a house (at WiFi frequencies) even with a single transmitter-receiver pair. In addition, there are RF communication standards such as Bluetooth Low Energy designed specifically to have low energy consumption.

This introductory Chapter is structured as follows: we give an outline of the thesis and its contributions in Section 1.1, followed by a description of the data used for all experiments in this thesis in Section 1.2.

[1] Home Care in Europe, accessed on 04/03/2023

1.1 Thesis Outline and Contributions

In this thesis we study unsupervised and semi-supervised learning of neural networks applied the problem of sensing humans indoors using radio frequency technology. We investigate active sensing, where extra hardware needs to be introduced to the monitoring area or the person needs to be wearing a wearable that actively communicates. We are concerned only with the case of a single person in some monitoring area, either a single room or a residential house. In the setting of active sensing, the extension to sensing multiple people is feasible albeit with reduced accuracy, however, we are limited by the availability of appropriate datasets. In the setting of passive sensing, the extra complication that would arise would be the need for tracking and identification, and considering these in the context of indoor localisation in real-world data is out of the scope of this work. Our data are time series of sensor readings, and we only consider active sensing with a single sensor, either frequency modulated continuous wave (FMCW) radar or a wearable communicating via bluetooth low energy (BLE) standard. Ambulation and position are the main two measurables, as learned functions of sensor signal, considered in this thesis. The machine learning libraries used in this thesis were: pytorch [8], scikit-learn [9], pyro [10] and flower (FLWR, [11]).

The use of unlabeled sensor streams is approached in a "day 0/day 1" scenario, day 0 being when the technician arrives to set up the system and collect labeled data, and day 1 being the day when the person being monitored is performing free living. When sensing humans indoors, we are likely interested on selecting a model, or hypothesis, before seeing new data, as fast prediction might be critical, for example in applications such as fall detection. We thus use the hypothesis learned on day 0 to make predictions on day 1, which is known as inductive learning. However, we are also interested in subsequently improving our hypothesis by using this new data. The timestep of a single day here is chosen as the minimum length of time to collect a sufficient amount of data to be confident of having chosen a better hypothesis, but it is important that this length of time is not too long for the performance with an outdated hypothesis to deteriorate significantly, which, as found by the authors in [12], is on the order of days for indoor localisation via fingerprinting. The thesis is structured as follows:

Chapter 1: Introduction. We outline the motivation and focus of this thesis, and introduce the datasets which are used for the machine learning experiments.

Chapter 2: Background. We introduce to the reader the mathematical formalism of statistical learning theory, focusing on inductive semi-supervised learning methods that are applicable to data with high-dimensional features.

Chapter 3: Ambulation Classification and Tracking. In this Chapter we explore ambulation classification with FMCW radar and unsupervised localisation of moving objects.

Chapter 4: Unlabeled Data in Fingerprinting Localisation. This Chapter focuses on the fingerprinting method of localisation in a residential scenario, and how the unlabeled stream can improve the performance of a fingerprint classifier through an approach using the time

dimension.

Chapter 5: Federated Learning for Fingerprinting Localisation. This Chapter describes experiments into the application of federated learning to the indoor localisation problem

Chapter 6: Conclusions. We reflect on the contributions and limitations of this work and discuss avenues for future research.

The contributions of the author to scientific literature have been the following:

1. Set-up, data collection and ground-truthing (ambulation and location) for the "UCL Ambulations Data" dataset [1] described in Section 1.2 below.

2. Parts of the work detailed in Chapter 4 have been published in conference proceedings [13].

3. The entire Chapter 5 is based on work accepted for publication in the 2023 IEEE International Conference on Consumer Electronics titled "Client Tuned Federated Learning for RSSI-based Indoor Localisation".

1.2 Experimental Data

Daily living involves several distinct timescales, the shortest that we can distinguish defined by the sampling period of the sensor. Indexing time at the sampling period of the sensor, we denote by $u_t = (x_t, w_t)$ the tuple of random variables that are the person's position x_t and pose w_t at time t. By position here we mean the tuple, either 2-dimensional or 3-dimensional, of co-ordinates in the frame of reference of the house, and by pose we mean a tuple, one element per keypoint, such as left shoulder or right knee, of co-ordinate tuples, with the co-ordinates relative to the person's position. The first timescale we wish to sense are ambulations which we denote by a_t, for example "stand up" or "fall down", taking on the order of a second. Next there are activities, which can take from tens of seconds, e.g. "washing hands", to hours, e.g. "working at desk". The longest timescale are small changes in the house that accumulate and have an observable effect of reducing the performance of a traditional supervised learner as quickly as in several days [12]. We investigate sensing ambulation and position, both at the time period of the sensor, to allow for cases such as incomplete ambulations. One possible causal model explaining the observed sensor signal at time t, s_t, is given in Figure 1.2.

Dataset 1: UCL Ambulations Data. This dataset [1] was collected by the OPERA Project team in a single-room office space. It involved a single subject performing one of six ambulations at several keypoints around the room. The ambulations recorded were: walk, bend, stand to sit, sit to stand, lie down to stand, stand to lie down. At any given time one of five different people performed each of these activities at different positions and orientations with respect to the radar. The duration of an ambulation is about 4 seconds, and there are about 100 examples of each ambulation, when summed over keypoint and over person. The modalities collected include: Lidar, FMCW, passive Wifi channel state information (CSI), continuous wave "Wifi radar" [14]. The FMCW radar used was the Texas Instruments AWR1642BOOST radar evaluation module. It is a complex-baseband fmcw radar with a linear array of 2 transmit antennae (tx) and 4 receive antennae (rx). The chirp configuration used was: $\{\text{slope}, \text{baseband sampling rate}, \text{chirp duration}\} = \{44\ \text{MHz}/\mu\text{s}, 3\ \text{MHz}, 100\ \mu\text{s}\}$. Figure 1.1 below shows the experimental layout.

Dataset 2: Residential Wearable RSSI. This dataset [15] was collected in four residential houses and contains several hours of recordings of time-aligned BLE RSSI, accelerometer readings and position annotations at meter squared precision, for a single person performing unscripted daily living activities. Communication with each AP happens roughly every 200 ms, and the position annotations are available with a period 40 ms. The default RSSI value for when an AP is has not communicated with the wearable is set to the minimum value of -108 dBm. The total duration of unscripted living is several hours per residence, recorded over the course of a single day, and there are also 'Fingerprint' sequences recorded for each residence, in which the person sequentially stands, for several seconds, on every single tile in their residence. More specifically, 'Fingerprint Rapid' involves standing briefly on each tile, while 'Fingerprint Floor' sequences involve the person standing for longer on each tile and turning to face each of the four directions.

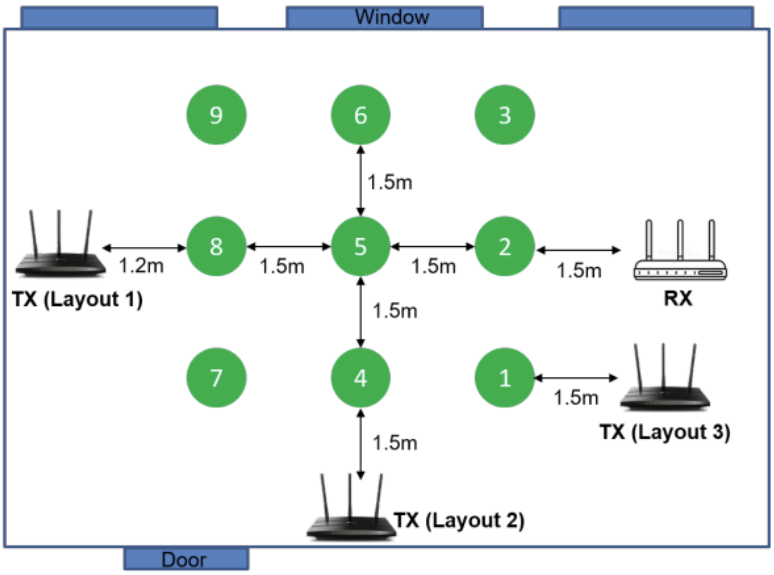


Figure 1.1: Experimental layout for the UCL Ambulations Data dataset described above, showing the dimensions of the monitoring area, the 9 keypoint locations, as well as the locations of the sensing devices. The FMCW radar and the lidar are not shown here, they are co-located with TX (Layout 3). The figure is taken from [1].

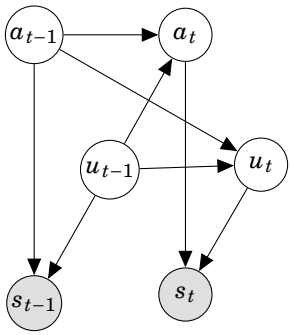


Figure 1.2: A possible causal graphical model describing the relationship between variables of interest in the ambulation, location and pose estimation problems. a_t is the unobserved ambulation, u_t are the unobserved position and pose, s_t is the observed sensor signal.

CHAPTER 2

BACKGROUND

Machine learning is a key ingredient of an indoor human sensing system and this chapter serves to introduce the reader to the learning methods explored in this thesis and the motivation behind them. Of the different machine learning tasks, in this thesis we mainly explore classification, therefore the exposition in this chapter will be centered on the classification problem.

The starting point of learning, in the language of statistical learning theory, is the learner's choice of a set of hypotheses $\mathscr{H}$. Each hypothesis $h \in \mathscr{H}$ is some mapping between two spaces called the feature and label spaces, and the learner seeks to either select a single hypothesis, or to produce a distribution over $\mathscr{H}$, given the data.

We find ourselves in the inductive learning setting, where the learner seeks to select a hypothesis prior to observing the "test" data on which the hypothesis will be evaluated. At the same time, the learner seeks to use the unlabeled stream to improve their hypothesis prior to observing new test data.

In Section 2.1 we introduce semi-supervised learning, the setting where a learner has access to both labeled and unlabeled data, and in particular the unsupervised domain adaptation problem. Classification with parametric and non-parametric models is then discussed in more detail.

In Section 2.2 we describe the problem of learning from high-dimensional features. Neural networks are introduced as one method of circumventing this problem, alongside with some classical dimensionality reduction techniques.

In Section 2.3 we go into more detail about consistency regularisation, giving a brief history of its sister term "contrastive learning" followed by recent theoretical results on the performance guarantees that can be made in the unsupervised domain adaptation setting.

Finally, in Section 2.4, we review Federated Learning, a method for distributed learning of neural networks, and introduce the Federated Averaging algorithm.

2.1 Semi-Supervised Classification

2.1.1 Learning with labeled and unlabeled data

We assume the learner has access to a labeled sample $D_L = \{(x_i, y_i)\}_{i=1...n}$ - the training set, and an unlabeled stream $D_U = (x_j)_{j \in J}$, J being a tuple of natural numbers indexing time that is not necessarily contiguous. $x \in \mathscr{X}$ is called the feature and $y \in \mathscr{Y}$ the label. The feature space $\mathscr{X}$ is a subset of R^d, and $|\mathscr{Y}| = r$ is the finite set of classes. S and T are the "source" and "target" sets, subsets of $\mathscr{X}$. The training set features are elements of S, while the unlabeled stream features are elements of T. A deterministic "ground truth" labeling function $y : \mathscr{X} \to \mathscr{Y}$ is assumed, where $\mathscr{Y}$ is the label space.

The quality of a hypothesis is measured by a loss function, also called cost or risk function, $L : \mathscr{X} \times \mathscr{Y} \times \mathscr{H} \to R^+$. The semi-supervised learner, in the simplest case, seeks to select a hypothesis h which minimises the target error,

$$\varepsilon_T(h) = P_{x \sim p_T}[h(x) \neq y(x)] \,,$$

over some feature distribution p_T on the target set, while only being able to evaluate the empirical error on the labeled sample, $\hat{\varepsilon}_S(h)$, as well as an empirical consistency loss on the set $U = S \cup T$. This specific setting is known as unsupervised domain adaptation, and, as the authors in [16] show, together with a some notion of distance on $\mathscr{X}$, provides sufficient structure to make it possible to state guarantees on $\varepsilon_T(h)$. The separation of S and T also makes explicit that the data on which we train and the data on which we test do not fall under the identically and independently distributed (iid) assumption.

Unlabeled data play an especially important role in the modern learning paradigm, when the hypothesis set is chosen such that there may be many hypotheses with low $\varepsilon_S(h)$, and regularisation is needed to reduce the number of plausible hypotheses. The idea is that a measure of compatibility [17] or consistency of a hypothesis on the data from U can help rule out a subset of these hypotheses, with two typical desiderata being low-density separation and local smoothness.

The former is the requirement that the decision boundary should pass through a region of lower data density, and the latter that the hypothesis h evaluated at some point in the feature space x' that is close to x, $h(x')$, should not differ significantly from $h(x)$. Note that a priori there is no notion of proximity in the feature space, and learning such a pairwise distance or similarity function will be discussed in Section 2.3, as that is a requirement to be able to make a connection between source and target error, as mentioned earlier.

2.1.2 Classification with parametric models

Classification models can be non-parametric or parametric, or a combination of the two. Non-parametric models, as indicated by the name, have no learnable parameters. Parametric models on the other hand are such models where the hypothesis set is parametrised by some parameter

vector $w \in W$, a subset of $\mathbb{R}^p$, with p the number of parameters. This makes it possible to search for the optimal hypothesis by using derivatives of the loss function with respect to w. A common choice of loss function for classification is the aforementioned classification error, also known as the 0/1 loss or accuracy, a metric that one would typically use to measure classification performance by default, though considerations such as the importance of false positives (FP) or false negatives (FN) may influence the choice of an alternative metric.

One immediate issue, if wanting to use derivatives to find the optimum of the loss function, is that the previously mentioned 0/1 loss is not a continuous function of w. A remedy is to choose a convex function, also known as surrogate loss, that upper bounds the original loss. Examples include the cross-entropy loss and the hinge loss. Only the former is used in this thesis, and can be written as, specifically with logarithm base 2 to be the correct upper bound to the 0/1 loss,

$$L_{xent}(x, y; f) = -\underset{y \sim p(y|x)}{E}\left[\log p_f(y|x)\right] = \log\left[1 + \sum_{c \neq y} e^{-(f_y(x) - f_c(x))}\right]$$

where f is a classifier outputting values that can be interpreted, after normalisation, as a predicted probability distribution over labels, $p_f(y|x) = \text{Softmax}\big(f(x)\big)_y$. The relation to h is $h(x) = \underset{y}{\text{argmax}}\ f_y(x)$ when the "argmax" decision rule is used. The second equality is, slightly abusing notation, because $p(y|x) = \delta_{y,y(x)}$ in the deterministic labeling case with $\delta_{a,b}$ the discrete delta function.

2.1.3 Classification with k nearest neighbours

The k nearest neighbour (kNN) classifier is the only non-parametric classification model used in this thesis, either on its own, or in combination with a learned distance function. kNN is often the algorithm of choice when we know nothing about the data, its benefits are: the 'learning stage' is the computation of pairwise distances, there are few hyperparameters - the number of neighbours and the distance function, and its predictions are interpretable because we know which labeled datapoints affected the decision as well as their relative influence. In addition to that, its error in-distribution is at most twice the Bayes' error in the limit of infinite data [18]. The former point of interpretability, however, is more applicable to data such as natural images, where we may interpret the decision by inspection of the neighbours, and less so to arbitrary sensor data.

The downsides of k nearest neighbours are that, as mentioned in the previous paragraph, a priori we do not have a distance function on the feature space, and that labeled examples must be stored to make predictions, which may make its use in memory-constrained applications impractical. This point becomes especially important in a continuous learning setting, with "representer point" methods, which we expand on in Section 2.2 when discussing dimensionality reduction, being a potential workaround.

The decision rule of the kNN classifier is

$$g(x) = \underset{y \in \mathcal{Y}}{\operatorname{argmax}} \sum_{x' \in kNN_{S,d}(x)} \omega(x', x) p(y|x')$$

where $kNN_{S,d}(x)$ are the k nearest neighbours of x in S, as ranked by the distance function $d(x, x')$, $p(y|x') = \mathbb{1}[y = y(x')]$ for typical "one-hot" labeled data, and some common choices for $\omega(x'|x)$ being

$$\omega(x'|x) = \text{const.} \quad \text{or} \quad \omega(x'|x) \propto 1/d(x', x) \quad \text{or} \quad \omega(x'|x) \propto e^{-d(x',x)}$$

which we refer to as uniform weights, inverse distance weights and exponential weights, respectively. Common choices for d are distances induced by the L_p-norm, $L_p(x) = (|x_1|^p + |x_2|^p + ... + |x_d|^p)^{1/p}$ with $p = 1$ or $p = 2$. Negative inner product is normally used for d when the representation, or embedding, space is the hypersphere. We discuss learning a parametric embedding function in Section 2.3 below.

Using L_p metrics for d in kNN as the standard first choice is not only unmotivated, but ill-advised because of the fact that the pairwise distance contrast vanishes with increasing feature dimension [19], and if we limit p to be a positive integer, L_1 would then be the optimal choice if we had no way of obtaining a distance function.

2.2 High-Dimensional Features and Neural Networks

2.2.1 Feature learning and dimension

One approach to choosing $\mathscr{H}$ could be to start with a simple hypothesis class and expand it until a hypothesis with sufficiently small empirical source error is found. One might, for example, start with a linear classifier, followed by a linear classifier on an expanded set of features consisting of all possible pairwise products of the features, and then by triple products and so on. A benefit of this approach is that we could select the "simplest" hypothesis (as in Vapnik's principle of structural risk minimisation [20]) by using a regularisation penalty on the magnitude of the weights, progressively strenghtening it while interpolating the training data, and then simply counting how many of the effective linear classifier's weights are nonzero. This makes sense if one considers that there is more flexibility in how much one or more of the features can change until we reach the decision boundary, hence such a classification is more confident.

A downside of the approach is that it may not be applicable due to too high computational complexity when the features are high-dimensional. However, that is not the biggest problem, but rather the fact that learning from high dimensional features is impossible in the first place [21] without further assumptions, known as the curse of dimensionality. One explanation for why modern learning methods work at all is the manifold hypothesis, that the samples of the ground-truth function to be learned are generated by a dimension-increasing mapping from some lower-dimensional manifold. The intrinsic dimension can be the number of non-zero fourier components of a bandlimited ground-truth function, or the rank of the adjacency matrix of the data similarity graph, introduced in Section 2.3 below. For the ImageNet [22] dataset of about 1M natural images, for example, the authors in [23] find that the manifold dimension is around 30, however, the assumption of a single intrinsic dimension for all data is only a convenience, and may not be necessary at all if we only use pairwise distances for classification.

The classical solution to the problem of high-dimensional features is to reduce the dimensionality of the data prior to classification. When the dimension is extremely high, to an extent that none of the following methods apply, there is always the method of random projections [24]. Otherwise, one can always start from the well-known linear dimension reduction method, principal component analysis (PCA, see [25] for a review), or one of the two classical non-linear methods: Multidimensional Scaling (MDS) and the related ISOMAP [26], where the former method solves an optimisation problem that maps input points to points in the lower-dimensional space such that pairwise distances are preserved as best as possible before and after the mapping, and the latter does the same but uses geodesic distances in the embedding space, computed on the nearest neighbour graph constructed using neighbourhood defined by a chosen distance function in the input space.

An issue with the above dimensionality reduction methods is that they can only represent global structure, as the authors show in [27], ISOMAP won't work even for the simple case of two

overlapping discs when they are parametrised on the manifold by some position θ, and one should really be "patching together the low-dimensional manifold". The authors of the latter paper have subsequently proposed the Hessian eigenmaps [28] that is akin to the locally linear embedding (LLE, [29]) but, as the authors show, does better the "patching together". LLE and related algorithms are of independent interest in their applications as "representer point" methods [30] - how good the local reconstruction is can help determine important datapoints.

Of the two most popular modern methods of dimensionality reduction, t-distributed stochastic neighbour embedding (tSNE, [31]) and uniform manifold approximation and projection (UMAP, [32]), the latter will be used in this thesis specifically for visualisation purposes. It essentially has only 1 hyperparameter, the number of neighbors, which creates for the user a choice of trade-off representing faithfully more of the local or the global structure, though UMAP is able to preserve well both (see Figure 4 of [32] for a visual comparison with several other dimensionality reduction techniques). The algorithm proceeds by first constructing the neighbourhood graph in ambient space, then constructing the neighbourhood graph in embedding, and finally optimising embedding graph to best represent the ambient space graph via "pairwise push-pull forces".

2.2.2 Neural networks and their properties

Neural network classifiers supersede the aforementioned procedure to choosing $\mathscr{H}$ by combining a learnable map $f : \mathscr{X} \to \mathscr{Z}$ from the feature space to some lower-dimensional representation space $\mathscr{Z}$, a subset of R^k, which is directly optimised together with a linear classifier on this representation by minimising a given loss function over the source dataset. It is observed that the optimisation process of neural networks via minibatch stochastic gradient descent (SGD), which we by default imply when mentioning neural networks, goes through both the aforementioned phases of finding a region in weight space with interpolating hypotheses, and further searches for a simpler one, a phenomenon termed double descent [33].

Two desirable properties of neural networks are the universal function approximation property [34], and their generalisation ability, in the sense of small difference between ε_S and $\hat{\varepsilon}_S$, or generalisation error, which can be attributed to a combination of the inductive bias in the architecture and stochastic gradient descent, with emphasis on stochastic. An explanation for the effect of the latter is that SGD is able to find flat minima, as defined by the eigenspectrum of the hessian of the loss function with respect to the parameters [35]. Ideally, we want to use a maximal non-diverging gradient step size combined with as many as possible random initialisations [36] to ensure that the parameter space is covered as well as possible, something that is particularly important to epistemic (model) uncertainty estimation, and the related problem of adversarial examples.

Stochasticity appears through learning with minibatches, and it has been shown [37] that the generalisation gap between neural networks trained with different minibatch sizes can be decreased or even eliminated by increasing the number of gradient steps with increasing

minibatch size. The latter paper's authors observed that training longer with a high initial validation rate, even after the validation error plateus, followed by a decreasing learning rate schedule for some more steps, leads to a comparable generalisation error with small minibatch GD. This is the aforementioned double descent phenomenon. Empirical support is also given by the authors in [38], after adjusting for optimisation difficulties, for the possibility of achieving equivalent generalisation error with large batch training as with small minibatches. On the other hand, the authors in [39] find that large batch training can lead to sharper minima in parameter space, while smaller minibatch training consistently finds flatter minima. A direct connection between the spectral norm of the hessian of the function with respect to parameters and generalisation error, when learning by SGD, can be seen in generalisation bounds derived using the concept of algorithmic stability [40].

Architectures of neural networks, while not the main interest in the thesis, are exploited to impose the appropriate inductive bias to the data at hand, when some understanding of the data allows it. We will use three kinds of architecture: the convolutional neural network (CNN, [41]) which is applicable when there is some sort of locality in the features, for example smaller recurring details in natural images, the recurrent neural network (RNN, [42]), and in particular its variant the gated recurrent unit (GRU, [43]) when the data have a time dimension, and, when the data is an arbitrary sensor stream, the feedforward residual network (ResNet, [44]).

Parameter count for neural networks will be chosen such that the training data can be easily overfitted, that is 0 source classification loss can be achieved. Weight magnitude is then a more appropriate quantity, rather than number of parameters, to measure complexity of neural networks [45]. As to the question of depth and width, there have been too many findings to faithfully give voice to them in the length of this paragraph, although depth separation [46] is one particularly interesting example, where the author shows that for an appropriate choice of ground-truth function, a depth-2 neural network cannot approximate it well, while a depth-3 neural network can. On the empirical side, the authors in [47] illustrate the empirical benefit of depth as a progressive simplification of the data manifold, which they have defined by computable quantities over the dataset. At the same time, the authors in [48] show specifically for ResNets that these don't need to be wide, in the limit of infinite depth, to be universal function approximators. Empirically, however, it is found that [49] the benefits of depth may not be fully utilised in a practical ResNet and that the expressivity of the neural net will rather scale with the number of neurons.

2.3 Consistency Regularisation

In the previous section, we saw that dimensionality reduction methods inevitably start with a distance function in ambient space, and it is also needed for the kNN classifier. Consistency regularisation can be considered a method for learning this distance function, and we start by introducing contrastive learning, a closely related term, before then transitioning to consistency regularisation and guarantees for unsupervised domain adaptation with use of unlabeled data. The term contrastive learning is first mentioned, to the best of the author's knowledge, in [50], in which the authors use the triplet loss for distance metric learning but use as negative examples the whole dataset, effectively taking the expectation over negatives in equation 2.1 below

$$L_{triplet}(x,x^+,x^-) = d_\theta(x,x^+) - d_\theta(x,x^-) \tag{2.1}$$

where we take $d_\theta(x,x') = d(f_\theta(x), f_\theta(x'))$ for the parametric embedding f_θ to be learned and a given distance function d in the embedding. Here x^+ refers to the "positive" or similar example to the "anchor" x, and x^- to the "negative" or dissimilar example to the anchor, and d is a distance function in the embedding space. If we consider the following modification, the key change being the change of the negative term,

$$L_{contr}(x,x^+) = \frac{1}{\tau} d(x,x^+) + \log E_{x^-} e^{-d(x,x^-)/\tau} \tag{2.2}$$

the loss becomes an alignment term and a uniformity, or entropy estimator, term [51]. Unless stated otherwise, contrastive loss will refer to this loss. The contrastive loss is, as the authors in [52] show, is "hardness-aware" - it has the same gradient as the basic triplet loss, but weighs more negative examples that are closer to the anchor. τ sets distance scale at which uniformity is enforced.

The representation space alignment and uniformity viewpoint has given way for the insight [53] that the contrastive loss is in fact the cross-entropy between the generating distribution of positive pairs and the parametric probability mass function effectively being learned, $p_\phi(x^+|x) \propto e^{-d_\phi(x,x^-)}$. The authors of this work have given population guarantees for inverting the data generating process under certain assumptions, the main one being the assumption that the latent variable for the anchor datum is sampled iid from a uniform distribution, and the positive example from the exponential.

Guarantees for the contrastive loss in a purely self-supervised setting, prior to having observed labeled data are desirable, as they can motivate choices for a practitioner. One line of research [54, 55] has sought to upper bound the logistic loss of subsequent classification using a linear classifier in the representation, and has resulted in bounds on the performance of the "mean classifier",

$$p(y=i|x) \propto \sum_{x' \in S_i} f(x) \cdot f(x') \tag{2.3}$$

unfortunately the assumption of conditional independence of positive pairs given their underlying "latent class" is too restrictive. For example, it does not hold when positive pairs are formed using proximity in time of sensor measurements or when positive pairs are formed using augmentations.

When semantics-preserving transformations, or augmentations, are used for positive examples of an anchor datum, for example masking out a small region of an image, we can refer to this semi-supervised learning procedure as consistency regularisation, a term which was mentioned in Section 2.1 in the context of using consistency conditions on unlabeled data to eliminate inconsistent hypotheses and this way reduce the number to choose from. The idea is by no means new, and has been applied to purely labeled data as early as in 1996 [56], where the authors fix function class $\mathscr{F}$, then use part of data to select a subset $\mathscr{F}_\epsilon$ of the function class with functions whose predictions are smooth on this part of the data, and finally use the remaining data to select f from $\mathscr{F}_\epsilon$ that also has low error on the remaining data.

The assumption of conditional independence is relaxed in the work of [57], where the authors give a population-level upper bound on the performance of a linear classifier in a representation learned by optimising a version of the contrastive loss with a different uniformity term, but one that can be shown to be same by Taylor expansion in the limit where the loss converges. They prove the fact that consistency regularisation is equivalent to learning a rank-k approximation of population augmentation graph adjacency matrix, and use that to obtain generalisation bounds of a linear classifier in the learned representation. In a follow-up publication [58], the authors generalise the statements to the unsupervised domain adaptation problem, they derive an upper bound on the performance of the mean classifier. In a simplified setting of gaussian clusters satisfying some reasonable assumptions, they show that in the embedding optimising the contrastive loss, the mean classifier "learned" using the source set can have good performance on the target set. This positive-pair graph is defined such that edge weights are simply given by $w(x,x') = P_+(x,x')$, the joint probability of sampling x and x' as a positive pair.

2.4 Federated Learning

Federated Learning aims to reduce the complications of centralised training, the main one being prohibitive data transmission (uplink) costs to a central processing server. However, more important in a healthcare context is the need to satisfy the privacy criterion that the clients' data must not leave their devices.

In federated learning, training takes place locally at each client using their private dataset and a centralised parameter server then aggregates the models of multiple clients. Various aggregation methods can be used at the server side before broadcasting the global model update to all the clients at each training and aggregation round. In [59] the authors introduce the FedAverage or FedAvg algorithm, described under Algorithm 1 of their paper, where the $k-th$ client's set of weights at round $t+1$, w_{t+1}^k are obtained by one or more gradient descent steps on the client's data, starting from the global weights at time t, w_t, which are obtained as in equation 2.4 below.

$$w_{t+1} = \sum_{k}^{K} \frac{n_k}{n} w_{t+1}^k \tag{2.4}$$

with K the total number of clients participating in this round, n_k the number of data of client k, $n = \sum_k n_k$. The authors of [59] call the specific case of a single gradient step at the client FedSGD, and they find that taking more gradient steps locally at each client decreases the number of rounds needed to reach a target accuracy of the global model, even for data that has been manually split amongst clients to be pathologically non-iid, as long as the clients all have a common initial set of weights.

Typically a 'plain' FL method such as FedAvg [59] does not necessarily give any individual client a better solution than the one obtained from individual learning [60]. As the authors of this latter paper show, even the simplest solution, local fine-tuning, makes it more desirable for the average client to participate in FL.

CHAPTER 3

AMBULATION CLASSIFICATION AND TRACKING

Ambulations are the next most fine-grained detail one can get about a person's daily living after pose and location, and include such critical applications as fall detection and unhealthy living pattern discovery. The first part of this chapter is therefore dedicated to exploring ambulation classification based on the FMCW radar modality, using the UCL Ambulations Dataset.

It is possible to localise a person either in depth (single antenna FMCW), depth and azimuth (linear antenna array), or depth, azimuth and inclination (2D antenna array), by looking at the velocity spectrum, which is the Fourier transform in the slow time dimension. It may be the case, however, that the slow-time dimension is already being used, for example for covariance matrix estimation in angular superresolution using the Capon estimator [61], making it impossible to compute the velocity spectrum.

The second part of this Chapter is therefore devoted to learning, in an unsupervised manner, to localise moving objects in image-like data, based on the simple prior that the movement is smooth and that the full image is a combination of one or more of these objects and a static background, as would be the case with an FMCW radar placed in the corner of a room, for example.

This Chapter is structured as follows: Section 3.1 details our ambulation classification experiments using the UCL Ambulations Dataset, and Section 3.2 describes the application of variational autoencoder (VAE, [62]) models to learn to track humans in image-like sensor data without position labels, partially inspired by the work of [63], attend, infer, repeat (AIR) and extending it to the non-static scenario.

Figure 3.1: Flow diagram of the proposed method, s_b is the baseband signal from the FMCW, a is the ambulation.

3.1 Ambulation Classification Using FMCW radar

Feature pre-processing. Fast Fourier Transform (FFT) is taken over each individual chirp, along the fast time direction (time domain samples), as well as over 256 consecutive chirps, along the slow time direction, with 224 chirps of overlap between two consecutive groups of 256. This creates a 256 by 255 snapshot in the range-velocity (RV) domain with the range being 0 to 10 m and the velocty –10 to 10 m/s. After beamforming by taking an FFT over the antennae direction, with the transmitted signal from the second transmitter forming an extra 4 virtual receivers, the power arriving from all angular directions is then simply summed, as there is only 1 person in the room at any time.

With this configuration, these range-velocity-angle (RVA) snapshots, each of which is a $(N_K, N_A, N_M) = (256, 8, 256)$-dimensional spectrogram, have a period of 6.4 ms, and an activity example is defined as being made up of 605 RVA snapshots, about 3.9 seconds. Because there is only ever a single person in the monitoring area, we sum over the angles in each spectrogram and use the 2-dimensional range-velocity (RV) snapshots. The radial distance spectrum is sampled in bins of width $\Delta r \simeq 4$ cm, the radial velocity spectrum in bins of width $\Delta v \simeq 7$ cm/s. The training set is made with about 450 examples in total (walking being the most represented, standing to lying down the least), and the validation set is made with about 80 examples.

It is desirable for the model to take as input the frequency spectrum (in our case it corresponds to power in radial distance, azimuth, and velocity coordinates) rather than the raw signal, as the former makes it possible to use local operators (convolutions), which are more parameter efficient.

Initial classification result. We find in initial experimentation that that a CNN classifier using as input the whole spectrum directly does not yield a classification rate above that of a classifier which always outputs the label of the class with the most training examples, while cutting out a small region of interest based simply on energy does yield a good result.

Thus, for each RV snapshot, a region of interest (ROI) of 32 by 56 pixels is made, about 1.25m by 4 m/s, which is enough to fully contain the person. The center of the ROI is chosen by where energy is maximum, and the ROI is smoothed over each activity example using a simple Bayesian smoother with a Markov model of ROI dynamics. An example of the spectrum and the ROI are shown in Figures 3.2 and 3.3, respectively. Figure 3.1 shows the flow of the method, with the ROI selection for a given frame depending on the ROI of the previous frame. In subsequent experimentation, the flow will remain the same, but ROI selection will be replaced with a learnable function.

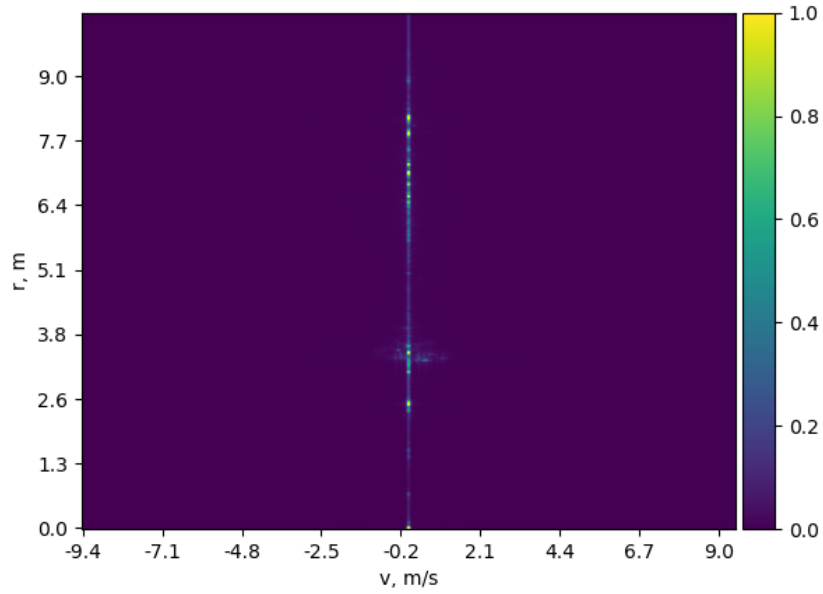


Figure 3.2: Range-Velocity amplitude spectrum of a snapshot from a single activity. The full spectrum is shown prior to removing the $v = 0$ bin.

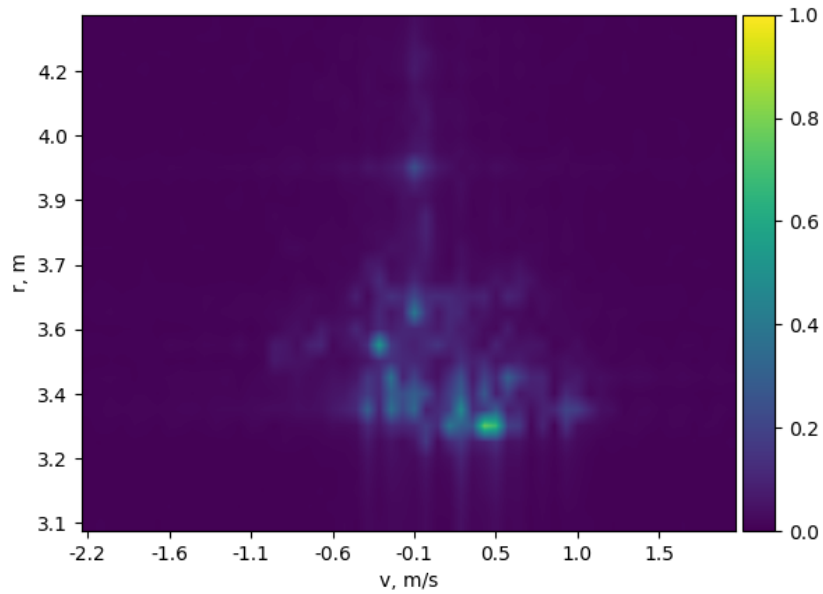


Figure 3.3: Region of interest extracted from the spectrum in Figure 3.2 above, as described in Section 3.1.

The neural network is a CNN followed by a GRU. The CNN processes the ROI to produce a lower-dimensional embedding of the spectrum. This embedding, together with the RV center

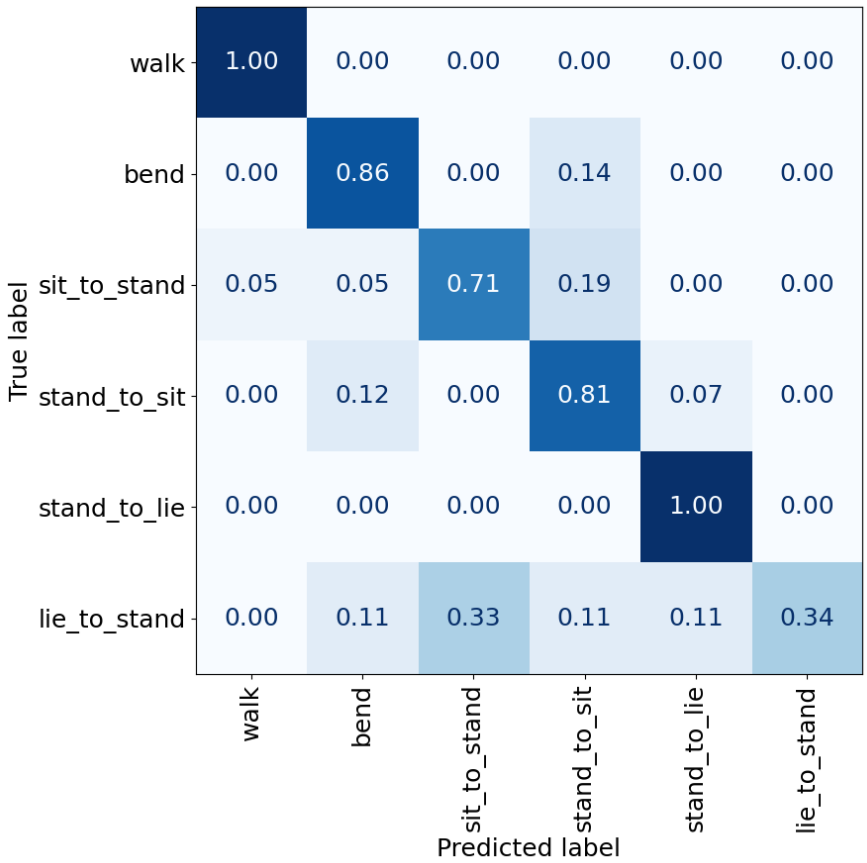


Figure 3.4: The confusion matrix on the validation dataset for the initial radar activity classification experiment as described in Section 3.1. The class decision rule here is simply the maximum of the output class probability estimates.

coordinates of the ROI, is then passed through the GRU for the 605 snapshots of each activity. Backpropagation through time is used to update the weights of the networks. The output is a softmax layer, and the class assignment is made by taking the maximum of this output layer.

The resulting confusion matrix is shown in Figure 3.4. It is clear that walking will be distinguished with ease given that the center of the ROI will vary most, and some of the expected confusion, such as that between sit to stand and bend, as well as that between stand to sit and stand to lie, is evident. This result motivates the removal of activities involving lying in subsequent classification experiments.

Second classification result on 4 activity classes. In the next iteration of experiment, changes are made to the machine learning algorithm to make the learning process more data-centric. It is expected, given the amount of data does not increase, that this may only be, at best, as good a result as that with manual selection of ROI + smoother.

The changes are - automation of ROI selection using the Spatial Transformer network (ST, [64]) and ROI tracking replaced by a recurrent neural network (RNN). In addition to that, the

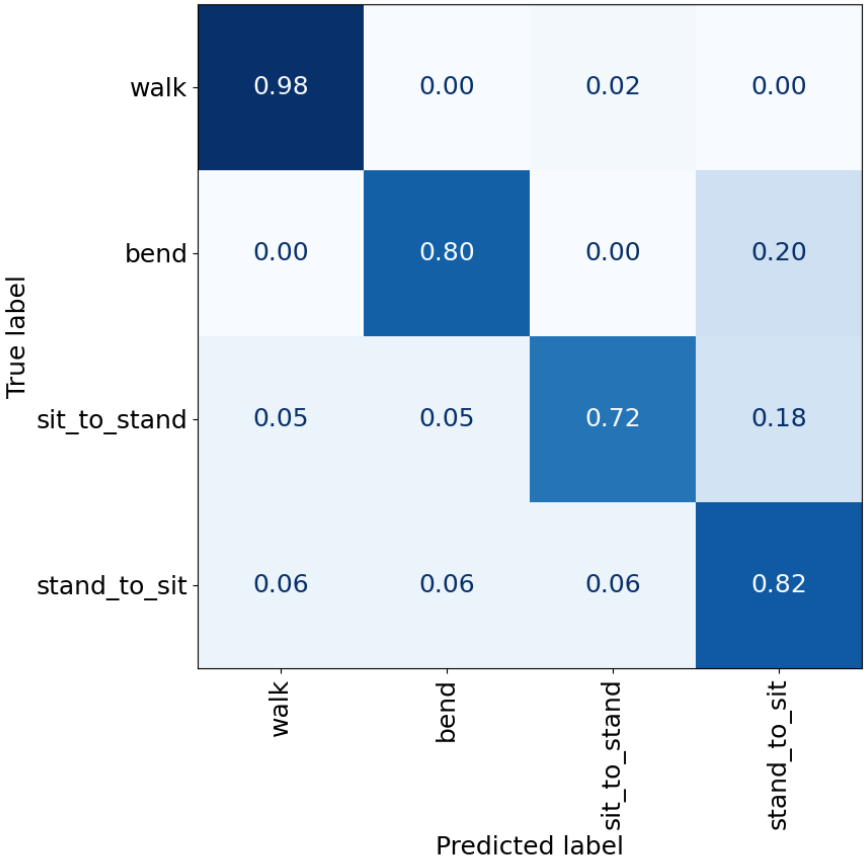


Figure 3.5: The confusion matrix on the validation dataset for the second iteration fully data-centric architecture described in Section 3.1.

treshold for decision is set at 0.5, fixing the minimum "confidence" of the classifier, and the set of activities has been reduced to the 4 excluding activities involving lying, as mentioned before.

The best performing classifier obtained an unweighted average F1 score of 0.77 at p = 0.5 treshold on the validation set. The confusion matrix is shown in Figure 3.5.

3.2 Unsupervised Localisation with Autoencoders

We hypothesize that it is possible to localize the person by inference on a generative model, which allows to include the inductive bias that the person is a finite size object with a variable pose and position added to a static noisy background. To test this hypothesis, we generate a synthetic dataset of grayscale image sequences which feature a single target with time-varying internal composition together with a static background and a small additive gaussian noise, described in more detail below. This well emulates the range-angle spectrum of a linear-array FMCW when there is a single strong reflector present.

Our primary interest is in the variational autoencoder (VAE) models, and the specific application is partially inspired by the previously mentioned Attend, Infer, Repeat model, which is a static, multi-target application of guided inference, the term used in the Pyro probabilistic programing framework, which is the library we use for all experiments detailed in this Section. Our justification for using VAE is fast inference, as compared to MCMC-based methods, as well as interest in implementing the Spatial Transformer as in AIR, followed by extending it to the dynamic setting.

We use variational inference and parametrise both the likelihood function (model) and the posterior probability density of the latent variables (guide) as neural networks, thus making computation of the latter at the inference stage fast, with the possibility of real-time application in mind.

The dataset. We observe a 2D sequence $\{X_t\}_{t=1...T}$ where $X_t \in R^{N_x \times N_y}$ and $(N_x, N_y) = (32, 32)$. t indexes the timestep, which is of length $\delta t = 4 \times 10^{-2}$. $T = 100$. Each image X_t is the sum of a static background, a moving target of time-varying composition and a small additive gaussian noise,

$$X_t = X_{bg} + \tau_t + n$$

The target size is $(N_x^\tau, N_y^\tau) = (3, 3)$. Along the diagonal of the target, its amplitude is a time-varying function,

$$\tau_{t,00} = cos(2\pi f t \delta t) \quad \tau_{t,11} = sin(4\pi f t \delta t) \quad \tau_{t,22} = sin(2\pi f t \delta t)$$

with $f = 2$, which emulates a time-varying pose. $n \sim \mathcal{N}(0, \sigma^{obs})$, $\sigma^{obs} = 1/SNR$. SNR is fixed to 100 unless otherwise stated. An example of the data can be seen in Figure 3.7.

The model and the prior. Both position $x_t = x_t(z_t)$ and pose $w_t = w_t(z_t)$ are deterministic functions of the single latent random variable z which contains position and pose information. The graphical models for the data generating model (decoder) and inference guide (encoder) are shown in Figure 3.6. It was initially found that if the guide does not use temporal information, convergence to a good optimum is possible but highly dependent on initialisation, and the model would rather minimize the size of the target.

When the size of the target is bounded between $[s_{min}, s_{max}]$, the learning process then leads to the minimization of target energy through modifying the weights of the function f_θ in the

model, given in equation 3.1. Adding a prior on the target energy did not help escape such bad optima either. The final prior, model and guide that were found to work on this single dynamic target case are detailed in equations 3.1 and 3.2 below.

z is assumed to take a random walk

$$z_0 \sim \mathcal{N}(0,1) \quad z_t \sim \mathcal{N}(z_{t-1}, \sigma_z)$$

with $\sigma_z = 0.1$ to encode that at any given timestep one doesn't expect these to be too dissimilar from the values at the previous timestep. We note that since the actual position of the target in the spectrogram x_t, see model below, is a nonlinear function of z_t, this is not necessarily enforced.

Data generating model (decoder).

$$\begin{aligned} w_t &= w_t(z_t) \\ \text{Target}_t &= f_\theta(w_t) \\ x_t, s_t &= g_\theta(z_t) \\ \tau_t &= \text{ST}\big(-x_t/s_t, 1/s_t, \text{Target}_t\big) \\ X_t &\sim \mathcal{N}(\tau_t + X_{bg}, \sigma^{obs}) \end{aligned} \tag{3.1}$$

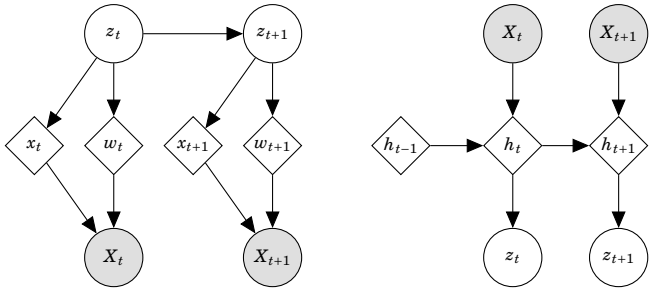
Inference guide (encoder).

$$\begin{aligned} \text{code}_t &= \text{enc}_\phi(X_t) \\ h_t &= \text{GRU}_\phi(code_t, h_{t-1}) \\ \mu[z_t], \sigma[z_t] &= v_\phi(h_t) \\ z_t &\sim \mathcal{N}(\mu[z_t], \sigma[z_t]) \end{aligned} \tag{3.2}$$

$x_t \in [-1,1]^2$ is the normalized target position in the spectrogram and $s_t \in [0,1]$ is the target scale, ranging from the target being infinitesimally small to occupying the whole spectrogram. ST is the spatial transformer module and X_{bg} is a variational parameter. f_θ is a neural network which takes as input the target pose w_t and outputs a $(N_x^f, N_y^f) = (8,8)$ image. g_θ is a neural network which takes as input z_t and outputs x_t and s_t confined to their respective domains.

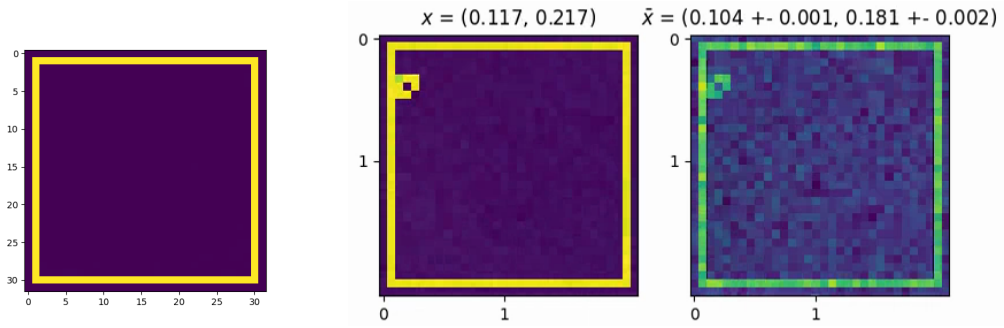
This model and guide pair is successful at learning the background and inferring the target location. This is demonstrated in Figure 3.7. The optimisation has not fully converged after 1000 steps, as the model learned $\sigma^{obs} = 0.09$ (the true value being 0.01). However, the position is correctly recovered and the reconstruction is visually accurate.

Multiple targets, single frame. Next, we investigate a direct extension to the case of multiple objects. There is no time dimension now, and the dataset consists of similar grayscale images as before, with each now having anywhere between 0 and 3 squares. Each square is either of size (3,3) or (5,5). The model and guide pair are shown in Figure 3.8. τ_i is a variable



(a) Model (b) Guide

Figure 3.6: Simplified model and guide pair for single-target dataset, with guide including an RNN.



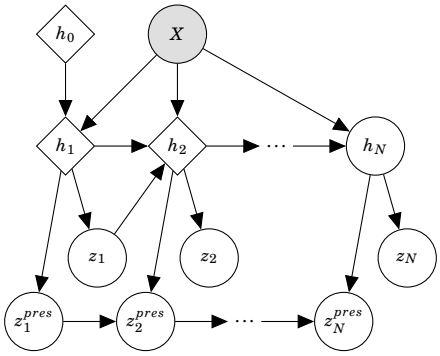
(a) Learned X_{BG} (b) Example reconstruction

Figure 3.7: In (b) we show a single frame from the synthetic grayscale sequences described in Section 3.2, and in (a) the background X_{bg}, a parameter learned by the model-guide pair described in equations 3.1 and 3.2. We can see in (b) that this model-guide pair is able to learn to localize the target in the single-target scenario.

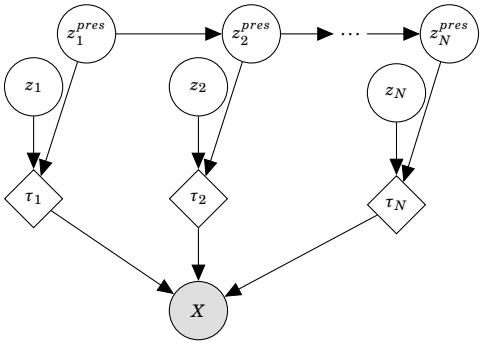
representing each target's small image, z_i^{pres} is an indicator random variable which is 1 if the i-th target is present and 0 otherwise.

About 10000 epochs are needed for convergence, however, it is always to a poor solution, with the model predicting the maximum of 5 targets on each data example. Some targets are repeated several times, such that when they are added the amplitude about sums up to the total. The authors in [63] propose a presence prior that is exponentially decreasing for each object, with the decrease factor annealed from a small value to 1 over the course of training, such that at the end of training the model assigns equal a-priori probability to all allowed numbers of targets. It is undesirable to have to make such specific modifications for every situation where we apply guided inference, hence we evaluate two alternatives with the hope that it is not a single fix that could make the multi-object detection module work in the static case.

The two alternatives are: an extra repulsion term in the loss to discourage the model from putting targets on top of each other, and masking out a detection after each inference step. Neither of these solved the problem, and the overall finding of this experiment was that the



(a) Guide



(b) Model

Figure 3.8: Initial model and guide pair for the multi-target single-frame dataset.

extent of hyperparameter tuning required to make such an inference procedure work is greatly exacerbated by the fact that one starts with an a priori unknown model, into which only some inductive bias (locality of the input data on a grid) is added, and only this inductive bias is insufficient, hence the tuning needed in some cases can be greater in terms of effort than simply collecting a sufficient quantity of labeled data.

CHAPTER 4

UNLABELED DATA IN FINGERPRINTING LOCALISATION

This chapter details experiments into the application of contrastive learning to room-level localisation by fingerprinting RSSI on the Residential Wearable RSSI dataset. Localisation is posed as a classification problem and a machine learning method is chosen over non-learning "direct" methods due to the fact that the latter are typically limited, for example to line of sight scenarios in the case of Ultra-wide band triangulation [65], or require extra information such as initial position, for example in dead-reckoning with bistatic WiFi [66] radar.

Fingerprinting is a method where a signal map of the environment is created prior to localization, and two issues with this localisation method need to be singled out. The first is that signal fingerprints need to be collected for each new deployment, and the second is that the quality of the fingerprints, as measured by classification accuracy, deteriorates noticeably on the timescale of days [12] due to environmental changes. This is not ideal, as it means that fingerprints would need to be collected weekly.

The first, labeled sample complexity, is proportional to hypothesis set size if it is finite, otherwise on a measure of its complexity, and the "day 0/day 1" approach of temporal contrastive learning used here is also a consistency regularisation method, for which we demonstrate an improvement in sample complexity by observing better testing performance at same training set size as compared to "day 0" classifiers.

The rest of the chapter is structured as follows: Section 4.1 presents the data splits and the experimental methodology - the classifiers that are tested and how their hyperparameters are selected, in both the "day 0"-only, and the "day 0/day 1" scenarios.

Then, in Section 4.2 we give the results of these experiments, as well as visualisations that it is hoped will help us understand better then interplay between different hyperparameters and the better performance of one method over another.

Section 4.3 concludes with a discussion of insights gained from this investigation.

4.1 Methodology

Our final performance metric is the test-set accuracy, and we select the hyperparameters for each learning algorithm by the greatest validation set accuracy. Because the validation set is a subset of the fingerprints collected, validation set metrics only ensure that the selected classifier has the best out-of-sample performance on the source data and has not obviously been fitted to the training sample.

RSSI is inexpensive and easy to collect, however, it is an unreliable signal for localization, as signal strength is strongly affected by the presence/absence of a line of sight which, in the case of a wearable transmitter, can simply amount to the wearer changing their orientation with respect to a receiver. We are thus also interested in how much a classifier confuses non-adjacent rooms, for which we use the room-level transition matrix (though generally we cannot assume that one will have access to it) $T_{y,y'} = P(Y_{t+1} = y' | Y_t = y)$. For a classifier g and labeled dataset D we compute the '1-step transition confusion score',

$$TC_1 = \frac{1}{n} \sum_{(x,y) \in D} \mathbb{1}[T_{g(x),y} = 0]$$

4.1.1 Data splits

Data in the Residential Wearable RSSI dataset has been collected in four separate residences, referred to as houses A, B, C and D. We do not use the data from House A as it does not have the same Fingerprint Floor and Fingerprint Rapid experiment sequences as the other 3 houses. We also do not use House B as the quantity of data collected in this house is much smaller compared to the other houses.

To allow us to use as feature short sequences of RSSI values rather than a single timestep while maintaining order and covering all rooms, we use the 'Fingerprint Floor' experiments of each house for training and 'Fingerprint Rapid' experiments for validation, these are two 'fingerprinting' sequences recorded in this dataset, as described in more detail in Section 1.2.

Of the remaining 'living' experiment sequences, we select 'living 5' for House D and 'living 2', 'living 8', 'living 9' and 'living 10' for House C as the respective test sets and the remaining 'living' experiments of each house form our unlabeled streams. For testing, when we use RSSI sequences of length 5, the datasets are formed by taking a step of 3 before the next datapoint that is a sequence of length 5, or roughly 50% overlap. The numbers of training, validation and testing data are given in Table 4.1

The tag ID corresponding to a location of a 0.2 s period is chosen as the modal tag ID in that period, and it is ensured that all tiles are visited in the training, validation and test sets (with the exception that the room "outside" in House C is not visited in the test set). It is also ensured there are no disallowed transitions in the data, and the features are normalised using pretraining data to have a mean of 0 and a variance of 1. For classification with sequences as features we also select the modal tile. The number of APs in both House C and House D is 11.

Table 4.1: Data split sizes

	Training	Validation	Testing
House C	16368	920	2355
House D	13747	605	1392

Table 4.2: Class distribution, House C

	pretrain	train	val	test
living_room	3535	4116	341	2902
kitchen	12304	3601	182	3395
stairs	314	933	28	133
hallway_upper	263	629	70	96
study	174	646	9	35
bedroom_1	273	2550	95	116
bathroom_toilet	414	944	43	258
bedroom_2	347	2655	119	180
outside	2081	294	33	0

Table 4.3: Class distribution, House D

	pretrain	train	val	test
hallway_lower	334	1248	56	41
stairs	366	939	45	109
living_area	4063	5144	207	2007
kitchen	5597	817	19	1496
bathroom_toilet	230	612	30	66
hallway_upper	438	1206	54	93
bedroom_1	2071	2491	144	182
bedroom_2	2413	1290	50	180

4.1.2 Classifiers

We start with the kNN classifier, with distances computed in the feature space, followed by neural network classifiers trained with the cross-entropy loss and finally we evaluate consistency regularisation on the source set only [67], using the labels, with the kNN classifier using the learned distance function. We then include the unlabeled 'living' stream, using temporal contrastive learning, where the positive examples are nearby in time to the anchor, in the "day 0/day 1" setting.

Training is always to stagnation, with a maximum number of stagnation epochs being 3, the reduction of learning rate by a factor of 10 after each stagnation, and the maximal number of stagnations allowed being 2. We always use the AdamW [68] optimiser, and we also try an exponentially decaying learning rate schedule, such that the learning rate at epoch t is equal

to the base learning rate times a factor of $e^{-t/\tau_{LR}}$. For the neural network models, we take the number of learnable parameters to be of the order of 10^5, finding that this is sufficient to achieve a loss value close to 0 for the cross-entropy loss.

kNN in the feature space. Fast evaluation of kNN allows us to choose the length of short RSSI sequences, T_s, by validation accuracy. We allow sequence length of up to 5 periods, or 1 second, $T_s \in \{1, ..., 5\}$. For a typical person's indoor speed of the order 1m/s, the loss of precision in taking sequences of length up to 5 RSSI periods is acceptably below the room-level localisation precision. We try both uniform and inverse distance weighting, in the latter case evaluating L_p metrics with either $p = 1$ or $p = 2$. For each of the 15 possible combinations of these hyperparameters, we try $k \in \{1, ..., k_{max}\}$. The maximum k value we choose to be the greatest possible feature dimension, which is $k_{max} = max(T_s) \times max(N_{AP}) = 55$, resulting in a total of $15 \times k_{max} = 825$ hyperparameter settings.

We use bootstrap to estimate the Spearman's rank correlation coefficient and its standard deviation between pairs formed from accuracy, inverse class-weighted accuracy, and the transition confusion score TC_1 of the kNN classifier. We use 10 resamplings of a fraction of 0.9 of all hyperparameter settings.

Having found that the best sequence length is 5, with details of all optimal, by validation accuracy, hyperparameters given in Section 4.2, we use this sequence length in all subsequent experiments.

Neural networks with cross-entropy loss. We optimise the cross-entropy loss, and also try the inverse class-weighted cross-entropy loss, with learning rates in $\{0.01, 0.001\}$ and weight decay factors in $\{0.1, 1\}$ selected by initial experimentation to be in a range where optimisation does not diverge. The minibatch size is 512 and $\tau_{LR} = 100$. With the given minibatch size and a maximum number of gradient steps of 1000, we train for a maximum of about 160 epochs. We evaluate two neural network models, the ResNet and the GRU.

The ResNet has hidden layers of equal width and ReLu activations. We try two different variants, one with 2 hidden layers of width 200, and one with 4 hidden layers of width 150. We also try a dropout factor of 0.1 on all hidden layers at training time. The weights are initialised such that the output units, assuming unit normal distributed inputs and sufficiently large width, are also unit normal distributed. For the GRU, we take the hidden dimension to have a size of 300, initialising the hidden state to zeros for each datapoint, and the initial weight values are sampled from $\mathcal{U}(-v, v)$ with $v = 1/\sqrt{\dim(h)}$. The total number of hyperparameter settings is thus 80, and we make 5 runs for each setting with a different random seed for the neural network weights.

Supervised Contrastive Learning. We optimise the contrastive loss using same-class positives and different-class negatives, a method proposed by the authors in [67], with $\tau \in \{0.1, 1\}$ and the number of negative examples $N_{neg} \in \{16, 128\}$. The learning rates are varied in $\{0.01, 0.001\}$ and weight decay factors in $\{0.1, 1\}$. $\tau_{LR} = 20$ and the minibatch size is 512. With

the given minibatch size and a maximum number of gradient steps of 1000, we train for a maximum of about 30 epochs. The smaller limits on gradient steps here are because of extra hyperparameter variations as compared to the cross-entropy training, and in the Results section it will be demonstrated that these settings still allow us to find hyperparameter combinations with good validation set performance.

We only use the GRU model as it was found to be the best model, by validation accuracy, for both House C and House D for the cross-entropy loss. We try representations on the hypersphere S^{d-1} and hypercube $[-1,1]^d$ with representation dimension $d \in \{10,55\}$, and the distance functions are the negative dot product for vectors on the hypersphere, and L_p distances with $p \in \{1,2\}$ for embeddings in the hypercube. With this number, 768, of hyperparameter settings, the training time is approximately 1 day on a GPU. In the embedding we use the kNN classifier using $e^{-d/\tau}$ weights with the same d and τ as used in the contrastive loss.

Unsupervised domain adaptation with Temporal Contrastive Learning. All hyperparam settings same as for SupCon above, except minibatch size of 256. The the positive examples are nearby in time to the anchor, and here we do not allow overlap, taking a step equal to the sequence length before selecting a positive example to the anchor. The anchors are formed with a sequence length of 5 and a step length of 3, the same way as the testing 'living' data is processed to form the test set. The numbers of anchors are 14282 and 12495 for House C and House D, respectively. With the given minibatch size and a maximum number of gradient steps of 1000, we train for a maximum of about 15 epochs.

Table 4.4: Validation set accuracies of best-performing, by validation accuracy, hyperparameter setting

	kNN (day 0)	NN xent (day 0)	SupCon (day 0)	Temporal CL (day 0/day 1)
House C	0.87	0.89	0.90	0.90
House D	0.84	0.87	0.88	0.91

Table 4.5: Test set accuracies of best-performing, by validation accuracy, hyperparameter setting

	kNN (day 0)	NN xent (day 0)	SupCon (day 0)	Temporal CL (day 0/day 1)
House C	0.94	0.93	0.93	0.94
House D	0.91	0.88	0.88	0.93

4.2 Results

Validation accuracies for each method and testing accuracies for each method, with the hyperparameter configuration selected by validation accuracy, are given in Table 4.4 and Table 4.5, respectively.

kNN. For both houses, the optimal RSSI sequence length was found to be 5, and the optimal neighbour weighing is inverse distance weighing with $p = 1$ of L_p metrics. The optimal k for House C is 17, and for House D it is 38. We show in Figure 4.3 the resulting validation accuracy when varying the sequence length, keeping the hyperparameters fixed to the best choice for each house. In Figure 4.1 we show the validation-set confusion matrices, and finally in Figure 4.2 we show the testing confusion matrices. We have also found interesting the plot of validation set TC_1 against k for the other hyperparameters kept fixed to the chosen for each house. This is given in Figure 4.4, comments on which we defer to the discussion in Section 4.3.

The values of Spearman's rank correlation for each pair of metrics are: SpearmanR(accuracy, TC_1) = -0.94, with standard deviation below 2 significant figures, SpearmanR(inverse class-weighted accuracy, TC_1) = -0.43 and SpearmanR(inverse class-weighted accuracy, accuracy) = 0.25, with standard deviations of 0.01 for both. The kNN classifier with hyperparameter choices giving better validation accuracy or inverse class-weighted validation accuracy will do so at the cost of making more non-adjacent room misclassifications. However, this does not definitively tell us whether we should be using inverse class-weighted validation accuracy for hyperparameter selection, as it correlates too weakly with accuracy, which is what we seek to maximise at testing time.

Neural networks with cross-entropy loss. We check that the standard deviation of the validation accuracy over the 5 random initialisations of the neural network weights is sufficiently small and use as classifier the argmax of the average logits over the 5 initialisations. For both houses, the GRU classifier optimising the inverse class-weighted cross-entropy loss with a learning rate of 0.01 was found to give the best validation set results. Test set confusion matrices

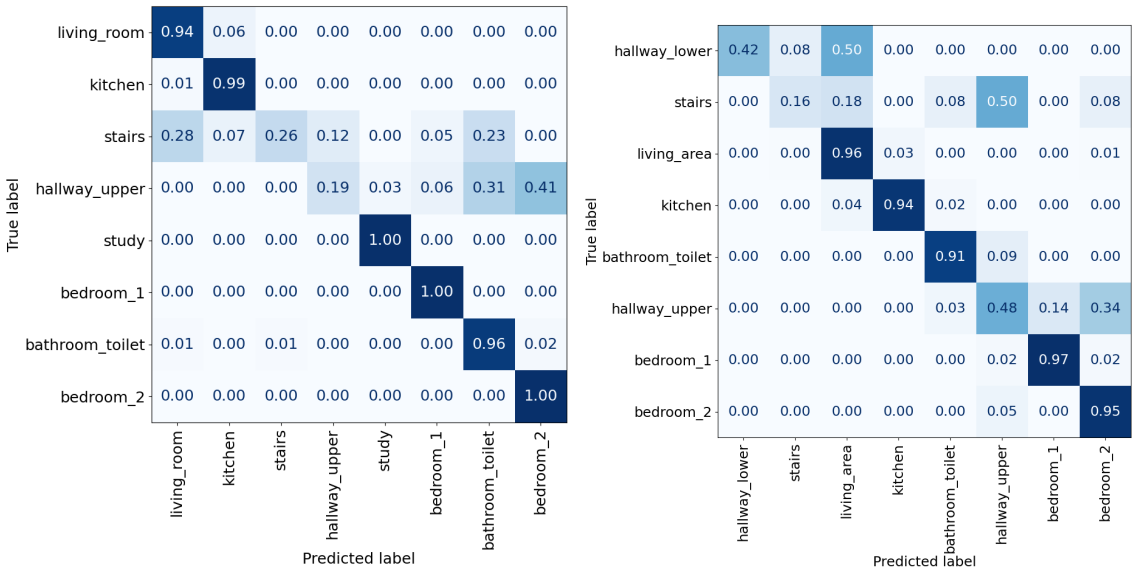
(a) House C (b) House D

Figure 4.1: kNN. Validation set confusion matrices, with hyperparameters selected by highest validation set accuracy. There is strong performance in the main rooms, such as living room, kitchen or bedroom, and it can be seen that signal propagation conditions and/or AP coverage in stairways and corridors are poorer. Note that there is confusion between living area and bedroom 2, which are rooms on separate floors of House D, and we would like the RSSI from these two locations to be well separated in an ideal feature space, as they are connected by several room transitions.

are given in Figure 4.5.

Supervised contrastive learning. For House C, the best $k = 1$, with the optimal representation on the hypercube with $d = 10$ and the distance metric being euclidean, and with $\tau = 0.1$, using $K = 128$ negative examples for optimisation of the contrastive loss. For House D, the best $k = 7$, with the representation on the hypersphere with $d = 55$, and with $\tau = 0.1$, using $K = 16$ negative examples for optimisation of the contrastive loss. In both cases the higher learning rate of 0.01 was optimal. Test set confusion matrices are shown in Figure 4.6.

Unsupervised domain adaptation with Temporal Contrastive Learning. For House C, the best k value was found to be 54, with the optimal representation selected on the hypersphere with $d = 55$, and with $\tau = 0.1$, using $K = 16$ negative examples for optimisation of the contrastive loss. For House D, the best $k = 54$ by validation accuracy, with the representation selected on the hypercube with $d = 55$ and the euclidean distance metric, and with $\tau = 0.1$ also, using $K = 128$ negative examples for optimisation of the contrastive loss. As with SupCon, the higher learning rate of 0.01 was optimal for both houses. We show the test set confusion matrices in Figure 4.8.



(a) House C (b) House D

Figure 4.2: kNN. Test set confusion matrices, with hyperparameters selected by highest validation set accuracy. Confusion in House D between floors, specifically between living area and bedroom 2, remains.

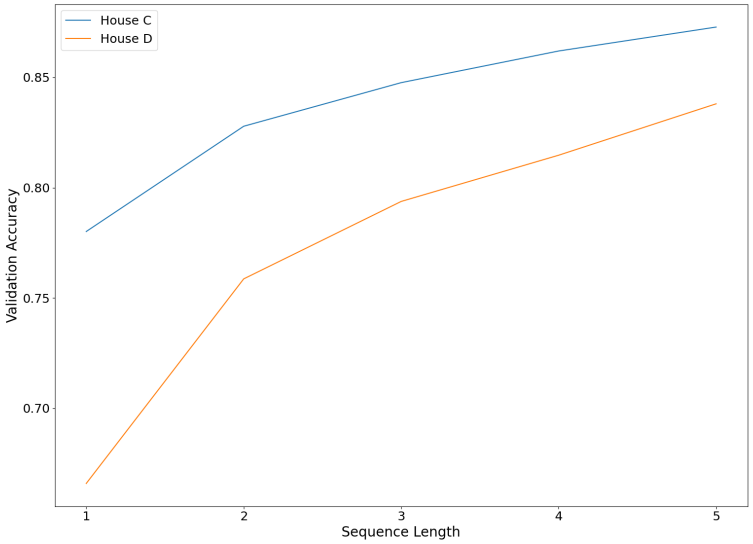


Figure 4.3: kNN. Sequence length versus validation set accuracy, with other hyperparameters chosen by highest validation accuracy. While there might be benefit of taking slightly longer sequences that 5 RSSI timesteps, that would also have the negative effect of reducing accuracy in transition rooms, such as corridors or stairways.

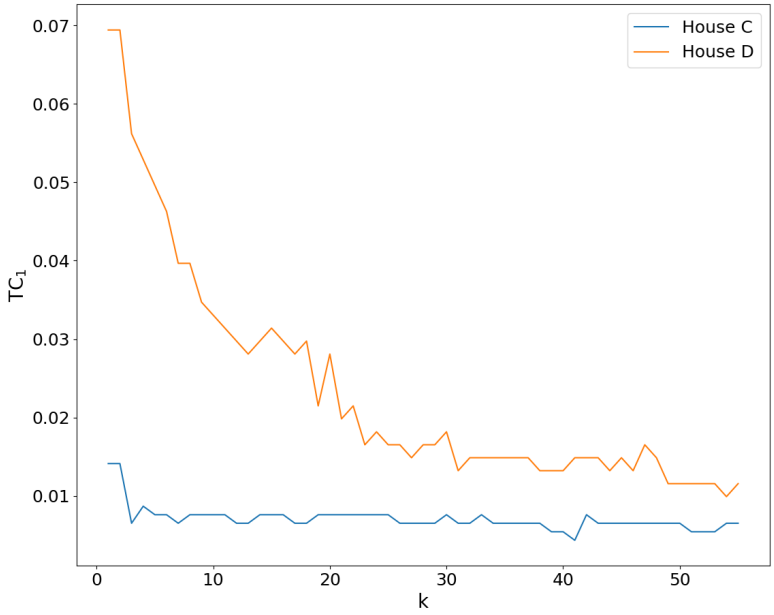
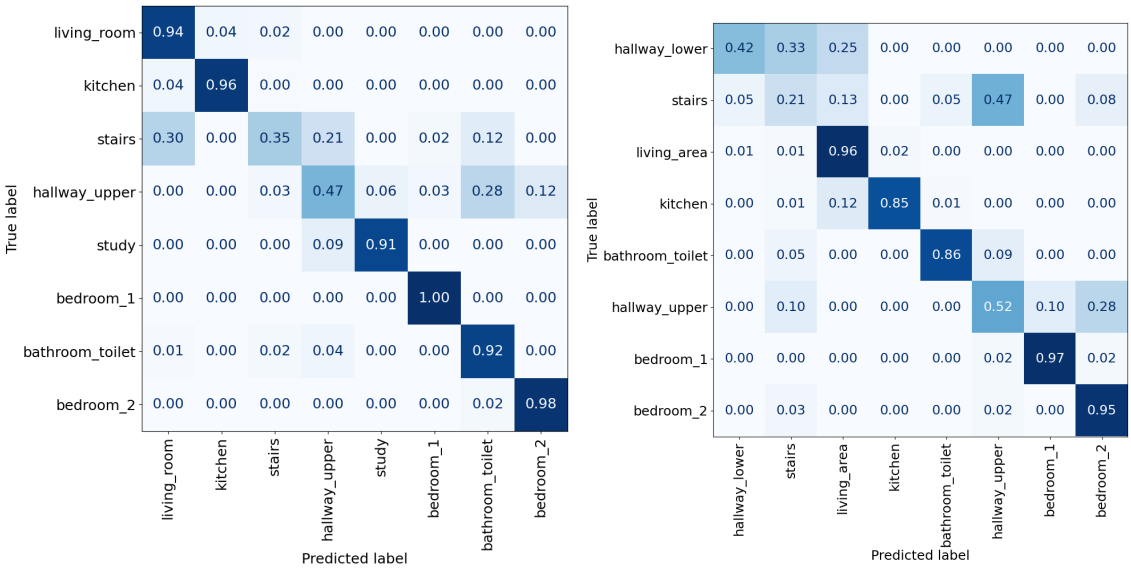


Figure 4.4: kNN. k versus TC_1, kNN, with other hyperparameters chosen by highest validation accuracy. The result for House D is also reflected in the accross-floor confusion seen in Figure 4.2.

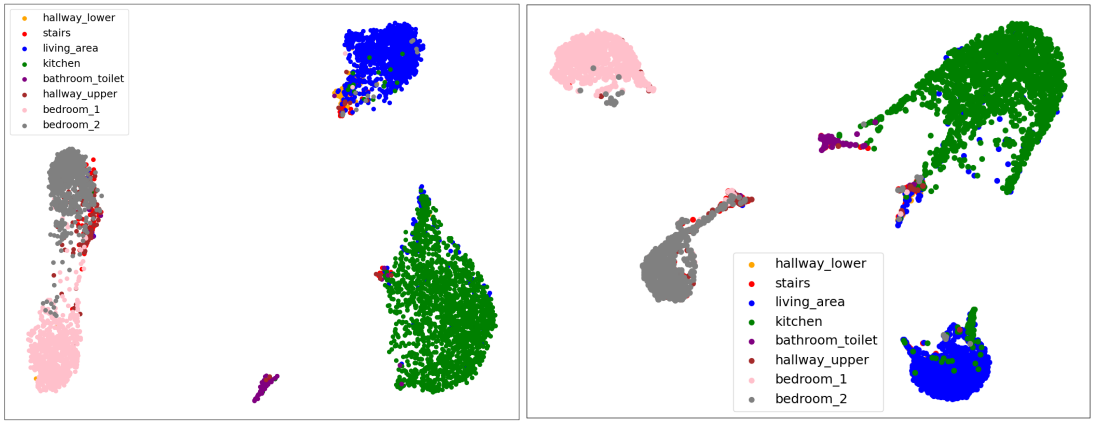
(a) House C (b) House D

Figure 4.5: Neural networks with cross-entropy loss. Test set confusion matrices. Compared to kNN in feature space, for House D the improvement that there is no longer confusion between floors (living area and bedroom 2) is promising, however, the final testing accuracy is worse than that of kNN in feature space in both houses. We also observe improved performance in transition rooms of both houses, likely due to the choice of inverse class-weighted cross-entropy loss, although this is at the cost of reduced performance in the main rooms.



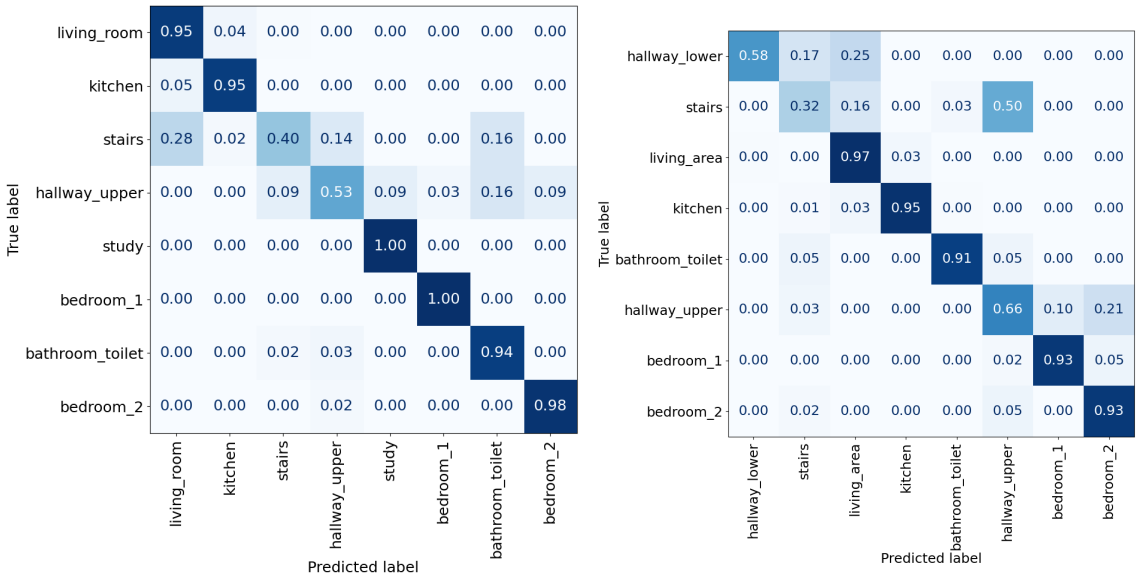
(a) House C (b) House D

Figure 4.6: Supervised contrastive learning, kNN classifier in embedding. Test set confusion matrices. The performance is difficult to distinguish from that of neural networks with cross-entropy shown in Figure 4.5, with the exception of being slightly worse in transition rooms. The low-dimensional UMAP visualisation comparing the embeddings learned by the two methods in Figure 4.7 may, or may not, give some explanation for that.



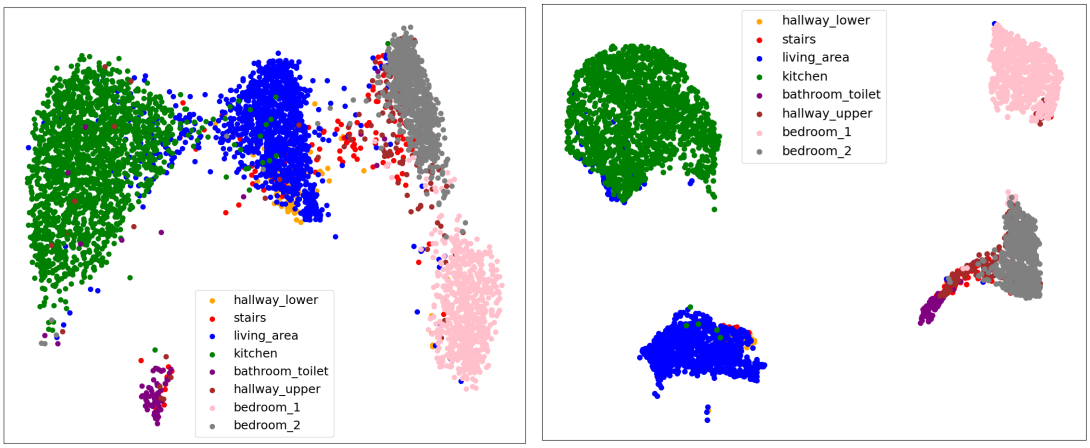
(a) NN xent (GRU hidden state) (b) SupCon embedding

Figure 4.7: House D. UMAP ($k = 50$, min dist = 0.1, L2 distance) visualisation of the unlabeled 'living' stream with ground truth labels for embeddings learned by neural networks optimising the cross-entropy loss versus the supervised contrastive learning method. Neither is clearly better, although bedroom 1 and bedroom 2 appear better connected in the NN xent embedding.



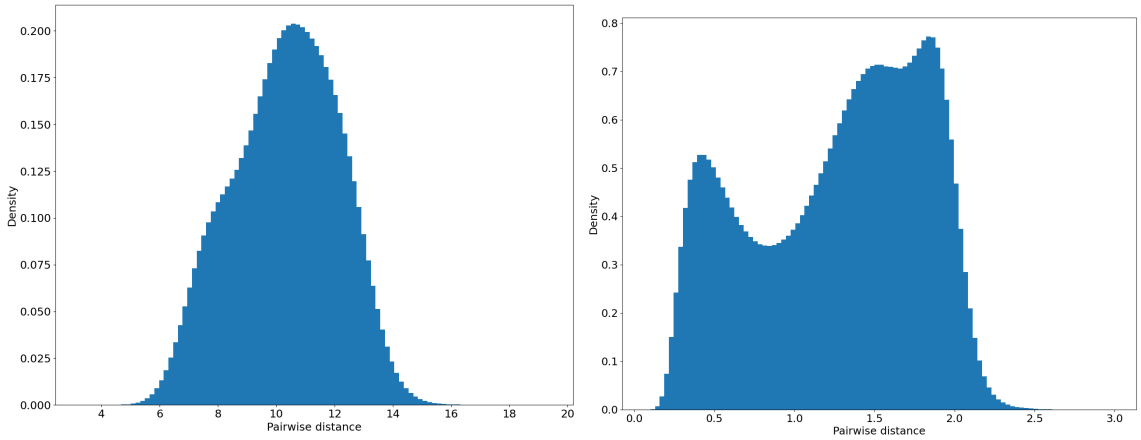
(a) House C (b) House D

Figure 4.8: Temporal contrastive learning. Test set confusion matrices. We can see that, in House D, there is no longer confusion between floors, specifically between living area and bedroom 2, unlike in the case of kNN in the feature space seen in Figure 4.2.



(a) Feature space (b) Embedding space

Figure 4.9: House D. UMAP ($k = 50$, min dist = 0.1, L2 distance) visualisation of the unlabeled 'living' stream with ground truth labels for the feature space and for embedding learned by temporal contrastive learning, chosen by highest kNN validation accuracy in the embedding. The clusters in embedding space are both cleaner and more compact, however, there is a lack of connectedness in the embedding space, possibly due to the value of $\tau = 0.1$ enforcing local, rather than global, uniformity. Nonetheless, this has not resulted in worse performance on transition rooms, but vice versa, as one can see comparing Figures 4.2 and 4.8.



(a) In feature space

(b) In embedding (temporal CL)

Figure 4.10: House D. Pairwise distance (L2) histograms of the data in our 'living' unlabeled stream. In both cases the dimension is 55, but clearly the distance contrast is greater in the embedding - one can see both the peak at distance of about 0.5 corresponding to the same room, and the second peak corresponding to the other rooms.

4.3 Discussion

Our first observation, from the test set accuracies in Table 4.5, is that kNN is the best of the "day 0"-only classifiers, or purely supervised learners. In addition to that, in the "day 0/day 1" case, when the embedding is learned by temporal contrastive learning on the unlabeled stream, the kNN classifier in the embedding is better, or at least as good as, in the feature space. Comparing, on House D, between kNN in feature space, Figure 4.2 (b), and kNN in the embedding learned by temporal contrastive learning, Figure 4.8 (b), we can see that there is no longer confusion between rooms on different floors in the latter.

The promising results on House D points out that the hypercube is preferred to the hypersphere which at least makes sense for low dimensions, as the spherical embedding "wraps around", whereas this is not true for the position space of a person. That the preferred dimension is the higher of the two evaluated, 55, is unexpected, given the discussion in Chapter 2 Section 2.1 about vanishing distance contrast rendering kNN effectively useless. At the same time, as discussed in Chapter 2 Section 2.3 it has been shown that for successful classification, albeit optimising a different variant of the contrastive loss, the dimension of the embedding must be at least twice greater than the number of classes. In fact, as the authors in [57] find, the target set performance bound only tightens with increasing dimension.

House D is also the more interesting of the two because, in the feature space, the L1 distance is much less trustworthy than in House C, as seen in Figure 4.4, and a k value of the order of 30 or more is necessary to find enough 'true' neighbors, in terms of position in the house, to offset this non-adjacent room confusion. As for the optimal τ value being the smaller of the two, $\tau = 0.1$, for temporal contrastive learning in House D, that indicates that local uniformity is either more important, which might be explained by the fact that clusters with low-density separation seen in the UMAP plot in Figure 4.9 are a desideratum for good classification, at least by a linear classifier. However, one would also like to see more connectedness, if one has reason to expect the embedding to reflect to some extent the position space of the person.

The main issue with optimising the contrastive loss is that we don't know what is the optimum value is and thus are not able to check whether optimisation has converged. However, there might be other ways to check the 'goodness' of the representation, for example the pairwise distance distribution, as seen in Figure 4.10. The clear contrast between two modes in the temporal contrastive learning embedding hints to some kind of parametrisation by a bimodal distrubition fit to the density plot where the parameters of best fit can then also distinguish representations by their quality, and, in fact, a weak overlap between the two modes when implemented as a regulariser has been shown to improve performance of contrastive learning [69].

CHAPTER 5

FEDERATED LEARNING FOR FINGERPRINTING LOCALISATION

This chapter describes experiments on the application of federated learning to the problem of indoor localisation via fingerprinting in a real-world multiple residential house scenario, using the Residential Wearable RSSI dataset. In applying FL to localisation in homes, we ask the question - how much 'common knowledge' can exist between them?

Federated learning can be seen as a regularisation technique, allowing to learn a feature extractor that focuses on residence-independent features in the signal. However, as discussed in Section 2.4 of Chapter 2, it is also expected that this will come at a cost of lower individual client level model performance as compared to pure individual learning if no extra per-client fine-tuning is performed.

We propose a similar method to the 'freeze-bone' fine-tuning described in [60], modifying it by having both input and output layers separate from the 'backbone' to adapt it to our scenario of fingerprinting localisation in a residential setting. A high-level overview of our setup and method is shown in Figure 5.1.

In order to effectively benchmark our proposed FL system, we perform extensive evaluation using a dataset of RSSI between wearable wrist watch and multiple access points located around the residence that act as receivers. We compare the performance of our proposed method against the model learned independently by the client on their own dataset, which we call individual learning, as well as the traditional, out-of-the-box deployment of FL [59].

The rest of the chapter is structured as follows: in Section 5.1 we motivate our method by giving a brief overview of previous applications of FL in fingerprinting localisation, after which we introduce the data splittings and experimental methodology, describing our tuned FL algorithm and how we compare it to standard individual learning.

Then, in Section 5.2, the results are presented, these indicate a significant reduction in the

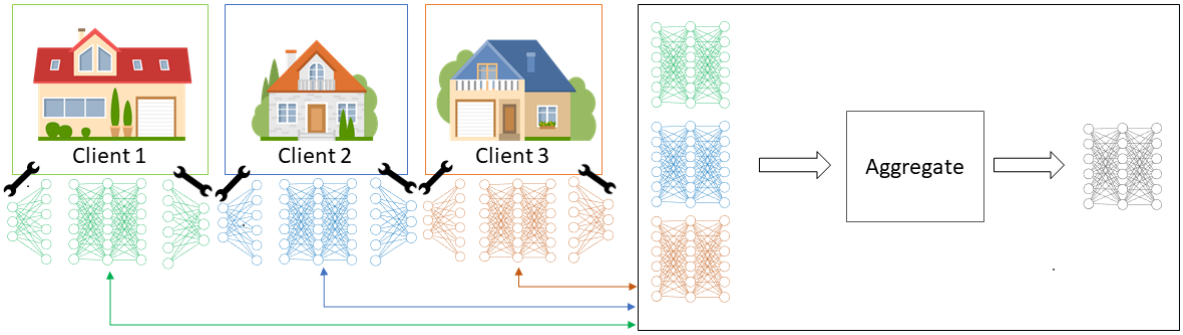


Figure 5.1: A diagrammatic overview of our method showing the shared backbone and the tuned client-specific parameters.

performance gap between a completely individual ML and a benchmark traditional FL approach.

Finally, Section 5.3 concludes this paper and sets possible directions for future work.

5.1 Motivation and Methodology

Prior art. Most of the research effort in application of FL to fingerprint-based indoor localisation has focused on data collected in the same environment but distributed over multiple clients [70, 71]. One common theme that is also relevant in our work is that of FL personalisation. Local fine-tuning [60] has been shown to be a promising method to close the aforementioned performance gap between individual learning and FL. The authors in [72] emphasize that the amount of local fine-tuning is ideally minimal.

In [71] the authors propose to address personalisation by using a 'Mixture of Experts', i.e., when a new datapoint arrives, the feature for it is a weighted combination of the feature predicted by a local model and by the current global model, with the weight itself a function of the datapoint. They use a parametric function of the client's own dataset for the weight. However, it is not clearly stated how the parameters are optimised. The authors of this paper also use self-labeling, which becomes especially important in a real FL scenario.

Another theme explored in FL for localisation is that of client heterogeneity and its detrimental impact on the result of FL. In [73] the authors compare FedAvg and FedAmp [74], finding that, in a scenario with clearly non-iid data where Clients mainly explore separate areas of the environment, FedAmp improves over FedAvg and gives closer results to those of the global model which is learned from all Clients' data combined. Client heterogeneity can also be addressed through different weighing schemes for combining the clients' weights at the server. In [75] the authors find that using an estimate of model reliability improves over FedAvg based on dataset weighing. A different weighing scheme, though only applicable to the scenario of data collected in the same localisation environment, is proposed by the authors in [76] - the area of space covered by the client's dataset.

Dataset splittings. We consider two basic classification scenarios: "Static", where the room-level location is predicted using RSSI measurement from the APs collected over a single period, and sequence classification, when k RSSI periods are observed before a class is decided on. "Static" classification allows us to use N-fold cross-validation (CV), where $N = 5$. We use an 80/20 training/validation split of the Fingerprint Floor sequences for each of the 3 houses. For classification based on a sequence we adopt the simpler approach of not maintaining a "hidden state" and instead classify based on the τ most recent RSSI samples. One issue is that we cannot splice the data for N-fold CV while maintaining order and covering all rooms in the training and validation datasets, hence we use Fingerprint Floor for training and Fingerprint Rapid for validation.

For testing, we use all of the Living sequences for each house. We ensure that all rooms are visited in all training, validation and testing sequences. We set the default RSSI value for an AP which missed communication during a period to -108 dBm and normalize the data to have zero mean and unit standard deviation for each AP. The dataset sizes are shown in Table 5.1.

Individual Learning. The optimal sequence length using the kNN classifier with the best k

Table 5.1: Dataset description

	Train/Val. ("static")	Train/Val. (sequence)	Test
House B	22262/5566	21925/689	6562
House C	15163/3791	15763/838	19473
House D	13053/3264	13165/561	19686

Table 5.2: Class distribution, House B

	train	val	test
hallway_lower	2111	35	1792
kitchen	2581	57	1088
living_room	2823	162	1060
dining_room	3073	61	1161
stairs	1429	42	341
bathroom	1575	41	269
hallway_upper	1139	31	208
bedroom_2	3056	82	199
bedroom_1	3589	160	342
toilet	549	18	102

Table 5.3: Class distribution, House C

	train	val	test
living_room	3946	326	5736
kitchen	3459	158	11464
stairs	912	20	366
hallway_upper	608	65	309
study	626	7	189
bedroom_1	2451	85	320
bathroom_toilet	908	38	590
bedroom_2	2572	109	469
outside	281	30	30

was selected from $k \in [1, 30]$ for each sequence length, shown in Figure 5.2. L1 distance was used as the metric in kNN as it gives slightly higher performance than Euclidean (L2) distance. For the linear classifier, the scikit-learn logistic regression implementation with default parameters was used. We find that, for the linear classifier, not using class weighing in the loss function gives slightly better results.

Having confirmed in initial experiments that the sequence classification case gives significantly better results, we focused on this case only. We use the GRU model, setting the hidden state dimension $\dim(h) = 300$, which gives about 3×10^5 learnable parameters. This is sufficient to achieve arbitrarily close to 0 training error on the sequence classification datasets of all houses.

Table 5.4: Class distribution, House D

	train	val	test
hallway_lower	1189	53	375
stairs	898	42	476
living_area	4938	196	6066
kitchen	769	16	7093
bathroom_toilet	584	27	296
hallway_upper	1156	50	533
bedroom_1	2393	132	2254
bedroom_2	1238	45	2593

The initial weight values are sampled from $\mathcal{U}(-v, v)$ with $v = 1/\sqrt{\dim(h)}$. The initial value of the hidden state is set to a vector of zeros.

Cross-entropy was used as our objective function with weights that were inversely proportional to the class distribution, as it was found to perform better when compared to unweighted cross-entropy.

For the optimisation, we use minibatch stochastic gradient descent (SGD) with the AdamW [77] optimiser and a minibatch size of 1024. While it has been shown that adaptive optimizers exhibit poorer generalization than "vanilla" SGD [78], they are significantly quicker to converge, which can be important in a computational budget-constrained setting. We set the maximum number of epochs to $N_{ep} = 50$ (this choice is informed a-posteriori by looking at the training curves with hyperparameter values described in the paragraphs below). We could also use a stagnation criterion to stop, for example if the training loss has not improved, or has changed relatively by less than 10^{-3} in 3 consecutive epochs.

We search for the optimal learning rate α and weight decay ω hyper-parameters in the respective sets $\{10^{-3}, 10^{-2}, 10^{-1}\}$ and $\{10^{-2}, 10^{-1}, 1\}$. The cartesian product of these is a set where training set loss does not diverge and by the end of the N_{ep} epochs and it has reduced to at least 1% of its value at initialisation. From each of these (α, ω) combinations, we make 9 training runs using a different seed for the NN weights as well as the minibatch sampler each run. The validation set loss is recorded at the end of N_{ep} epochs of each run.

Federated Learning. Unlike fitting a static dataset, FL is a continuous process in which, in general, a random subsample of clients may participate in any round, and new clients may join at any time. We explore an idealised setting where over a small number of rounds all 3 Clients are participating in each round and the initial parameters of the model are, at least from these 3 clients' perspective, random in the sense of not necessarily having a small initial loss value on the individual training sets.

The extra complication of FL applied to this scenario is that each house can have a different number of APs, hence the input dimension may differ amongst the clients, and each house may also have a different number of rooms. We address this by using, for each house, individual

linear input projection and classification parametric functions, which we subsequently call the **projector** and the **classifier** in short.

The shared intermediate feature extractor we call the **backbone**. The classifier and projector together we refer to as the **private** parameters of the model. The full function computed by this model on an input x is given by (Classifier $\circ$ Backbone $\circ$ Projector)(x). We use linear models for both projector and classifier, the former taking the N_{AP} RSSI values (for a single timestep) to an intermediate vector of dimension 100, and the latter mapping a 100-dimensional vector to N_C class scores. The backbone is a GRU with $\dim(h) = 300$.

We use the FLWR library to implement FL choosing the FedAdam [79] optimizer with learning rate η on the server for faster convergence. On the clients we use full batch GD with (α, ω) = (0.01, 0.1) for one epoch of training per fit round. This choice, together with $\eta = 10^{-3}$, gives the 'best-behaved' learning in terms of minimizing oscillations in the validation loss during the evaluation stage of each round.

For client training during the fitting stage of each round, the answer to the question of how long to train lies between two extremes: at one end of the spectrum, we may want an arbitrarily small step on the client (which is always guaranteed to improve its own loss) and to see at least n data examples with m per minibatch (limited by memory), in which case one can use full batch GD.

At the other end of the spectrum, one may want each client to converge, according to some criterion, to a local optimum of their training set loss during each fit round. The convergence criterion could, for example, be stagnation. We find experimentally that this choice leads to significantly longer training times. We therefore always use full batch GD for the fit round.

We introduce personalisation by allowing each house to fine-tune their private weights only during the evaluation round. Each Client computes the validation-set error on the most recent downloaded global model, does fine-tuning of their projector and classifier for N_{tune} epochs, computing the validation set error after each epoch.

If the best of those epochs' validation error is lower than the previous best, the client keeps the new tuned weights, while if it is higher, the client reverts to its private weights at the start of the fit round. This is summarised in Algorithm 1. We use in the tuning round the AdamW optimiser with (α, ω) = (0.01, 0.1) and a minibatch size of 1024. We limit the self-tuning to 5 epochs.

Algorithm 1 Tuned FL

Begin round t
for client $c \in C(t)$ **do**
  $\overline{w}_B(t-1)$ = DownloadWeights()
  if c not initialised **then**
    $L_{\text{min}} \leftarrow$ max_value(float)
    $w_P(t-1)$ = InitPrivateWeights(seed)
    $w_{\text{test}} \leftarrow \left(\overline{w}_B(t-1), w_P(t-1)\right)$
  end if
  $w_B(t), w_P(t)$ = Fit$\left(\overline{w}_B(t-1), w_P(t-1)\right)$
  UploadWeights($w_B(t)$)

  $\overline{w}_B(t)$ = DownloadWeights()
  DisableBackboneGradient()
  $w_P(t), L_{\text{val}}(t)$ = Tune($\overline{w}_B^i$, w_P^i)
  EnableBackboneGradient()
  if $L_{\text{val}}(t) < L_{\text{min}}$ **then**
    $w_{\text{test}} = \left(\overline{w}_B(t), w_P(t)\right)$
    $L_{\text{min}} = L_{\text{val}}(t)$
  else
    $w_P(t) \leftarrow w_P(t-1)$
  end if
end for

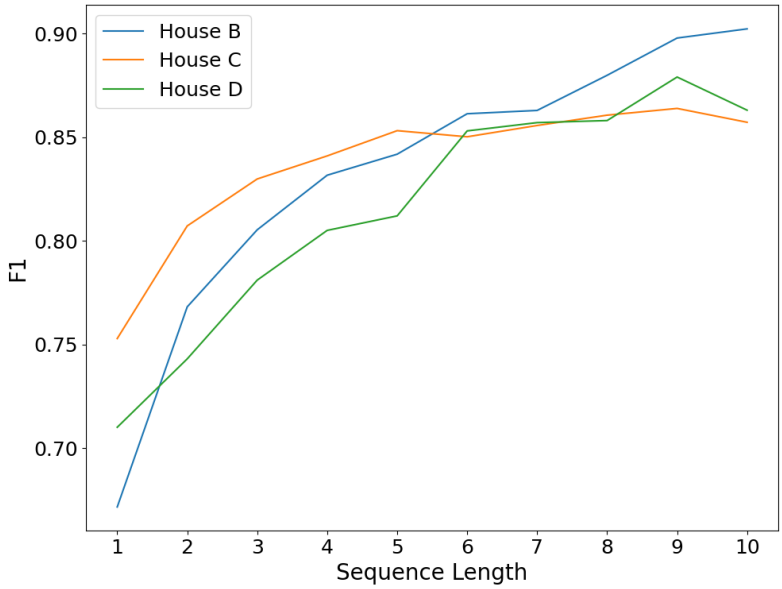


Figure 5.2: Validation F1 score for sequence lengths $\tau \in [1, 10]$ for the kNN classifier trained on the sequential datasets of each individual house. The best k for each sequence length is selected from $k \in [1, 30]$.

5.2 Results

Unless stated otherwise, we use the validation set loss as the validation metric. The testing metric is the test set F1 score, calculated for each label and averaged with weights proportional to the number of true instances for each label.

To measure overall performance over the 3 houses, a metric is averaged with weights proportional to the number of training/validation/testing data for each house, also referred to as federated averaging or FedAvg.

Individual Learning. The result of the kNN classifier on the sequential datasets, with varying sequence lengths, is shown in Figure 5.2. There is little improvement beyond $k = 10$ and for most sequence lengths the optimal number of nearest neighbours is around $k = 20$. We fix the sequence length to be $\tau = 6$ and use it in what follows as there is not significant improvement in overall validation F1 score for longer lengths, while the risk of misclassification, especially in transition locations such as short corridors or stairs, increases.

For the GRU, we find the best set of hyper-parameters to be $(\alpha, \omega) = (10^{-2}, 1)$ as this combination minimizes the validation loss and, more importantly, an estimate of the variance of the validation loss over realizations of weights at initialisation and minibatch example indices. The preference for a higher learning rate, when combined with learning rate reduction at the stagnation criterion described in Section 5.1, can be explained by the optimisation being more

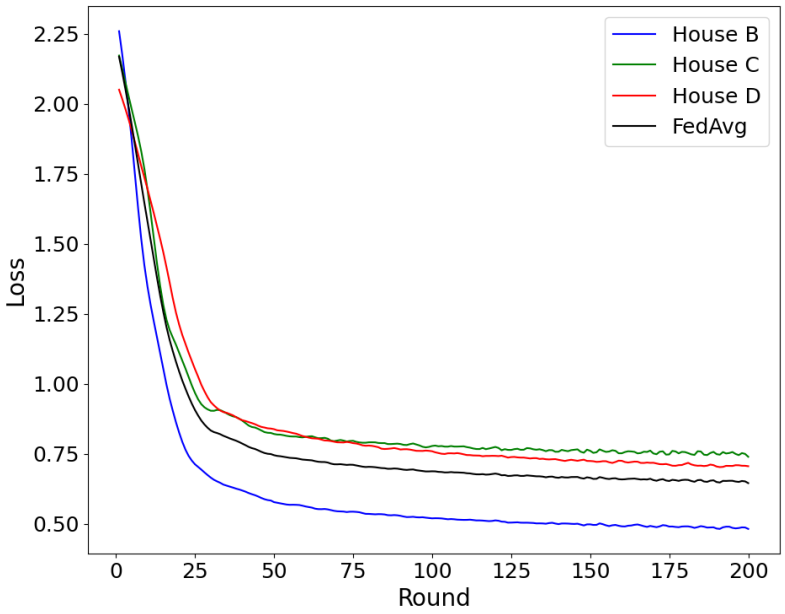


Figure 5.3: Validation set losses for the individual houses and the Federated average in the evaluation step of each round, benchmark FL. The errors, estimated as the standard deviation over 5 different initialisations of the backbone weights at the first round, are too small to be visible in the plot.

Table 5.5: Average acceptance rates of Algorithm 1 over 50 rounds.

House B	House C	House D	Overall
0.20	0.18	0.34	0.24

likely to escape to a better, at least in terms of training loss, part of the weight space starting from any weight initialization.

Federated Learning. The training curve for the empirically selected optimal hyperparameters (α, ω, η) = (10^{-2}, 0.1, 10^{-3}) is shown in Figure 5.3. The FedAvg of the validation error reaches 0.81 at 100 rounds and stagnates until the end of the 200 rounds. In tuned FL, stagnation was observed at circa 15 rounds, equivalent to 90 rounds of our benchmark FL in terms of computational budget. The federated average validation F1 score of 0.87 is achieved at the end of 50 rounds. The training curves for benchmark FL and tuned FL are illustrated in Figure 5.4.

We find that the acceptance rate of Algorithm 1 decays as training progresses. The overall acceptance rate as well as individual per-house rates over 50 rounds of tuned FL is given in Table 5.5. Finally, we compute the test-set scores for individual learning, FL and tuned FL. These are given in Table 5.6. The anomalously low scores of house B are due to low scores of about 0.5 on the sequence 'living 4' of this house.

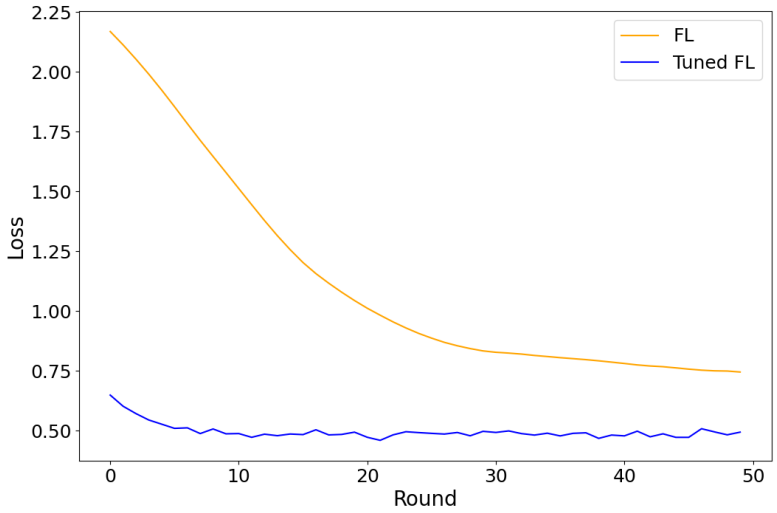
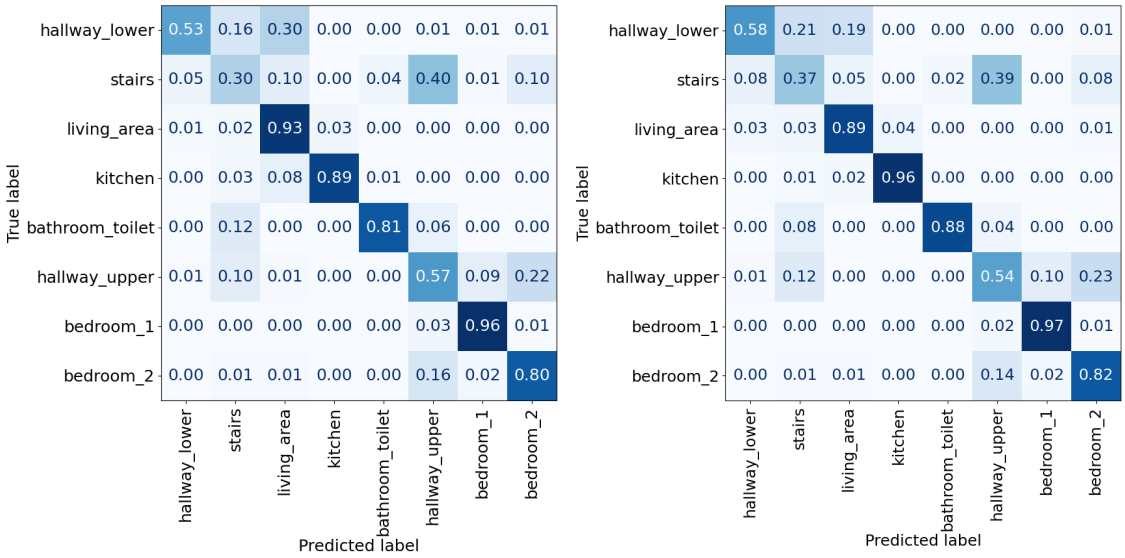


Figure 5.4: FedAvg Validation loss curves for FL and tuned FL. The errors, estimated as the standard deviation over 5 different initialisations of the backbone weights at the first round, are too small to be visible in the plot.

Table 5.6: House-averaged test-set F1 scores.

	House B	House C	House D	Overall
Individual learning (GRU)	0.76	0.90	0.87	0.87
FL	0.75	0.80	0.86	0.82
Tuned FL	0.76	0.84	0.90	0.85

For House D, for which we have the largest test set, we see an improvement of tuned FL over individual learning. The confusion matrices are given in Figure 5.5. It is evident that tuned FL has improved accuracy, as compared to FL, on most rooms, even though the final F1 score is most dependent on the main rooms where the person spends most time, such as the living room, kitchen and bedrooms.



(a) Individual learning

(b) Tuned FL

Figure 5.5: House D test-set confusion matrices for individual and federated learning.

5.3 Discussion

We begin by discussing two issues with our experimental method relating to the limitations inherent in the dataset having too small a number of residences and not representing faithfully the online/streaming aspect of the data that would be collected in a residential deployment. The first issue is that the initialisation of backbone weights sent by the server in the first round is randomly sampled and thus carries no 'knowledge' about the problem. The second is the issue of using the validation set for model selection in the self-tuning round.

In a real-world IoT FL deployment, due to the non-iid nature of the data, it is entirely possible that for a new FL client joining the network at round t, the global weights that it would be sent, w_t, not give better performance on their held-out validation data than a random set of weights. What we have shown is that, in the case where all clients start with a common set of randomly initialised weights at round 0, limited local fine-tuning leads to a better performance for all of the individual clients on their validation data as compared to FedAvg. At the same time, the aggregated model remains useful, which is also supported by the fact that the clients reach good performance on their individual data with limited fine-tuning.

As for the second issue of using validation error for model selection, it is clear that given a large enough number of rounds, if the amount of labelled data available is constant, the client will inevitably overfit their validation set. It is therefore paramount to supplement any real deployment with the ability to annotate new data locally. Active learning can be one approach, but would potentially require more hardware, possibly in the form of proximity sensors or some tags. Self-labeling is another alternative, explored by several works mentioned in Section 5.1. We have assumed that the timescales here in terms of the numbers of rounds here are short enough that validation set overfitting is of no concern.

We did not try a nonlinear projector and nonlinear classifier for the client's private functions as we sought to show that the representation power in the the backbone (weights) learned in FL is sufficient to obtain a stronger result on each individual house than if individual learning with the best linear classifier had been used. Even in the linear projection case, a permutation of the inputs can be learned.

CHAPTER 6

CONCLUSIONS

The main aims for this thesis were to investigate problems related to the use of machine learning in indoor sensing, in particular in ambulation detection and in indoor localisation, and to explore how unlabeled data can reduce the labeled sample complexity of learning methods in indoor sensing, as well as how unlabeled data can be used to ameliorate the problem of gradual worsening of classification performance over time when a purely supervised learner uses an initial data collection to learn a hypothesis that is subsequently used for new data.

In addition to that, in this thesis we explored whether localisation can be achieved in an unsupervised manner in generative models by making use of the prior that location at nearby times cannot vary significantly. We also investigated, in a purely supervised setting, whether it is feasible to use federated learning to share "knowledge" between houses without sacrificing privacy.

We now give a brief review of each of the three experimental chapters, followed by asking the remaining questions as well as proposing future research directions in Section 6.1.

Chapter 3 In the first part of this chapter, we looked into ambulation classification using a linear antenna array FMCW radar modality. We proposed a neural network that was shown to work well in place of manual feature engineering of the input features and demonstrated good classification performance for 4 key ambulation classes: walk, bend, sit to stand and stand to sit. In the second part of this chapter we investigated the application of variational autoencoders, or guided variational inference, to the unsupervised learning of localisation of moving targets in image-like data. We found that, for the case of a single moving target, the goal is achievable with the correct choice of inference guide, or encoder, in particular one that has temporal structure. We also found that the multi-target case, even just the static scenario, is much more difficult and the necessity of finding 'hacks' to make the inference work hindered further progress.

Chapter 4. In this chapter, we focused on the indoor localisation problem and in particular the fingerprinting method of localisation. We sought to use unlabeled data by relying on the temporal structure, a method known as temporal contrastive learning, to improve on classifiers that are purely supervised learners. We compared 3 supervised-only learners, kNN, neural networks minimising with the cross-entropy loss and supervised contrastive learning, finding that kNN with distances measured in feature space is the superior of the three. We then evaluated whether the kNN classifier will do better in the embedding space learned by temporal contrastive learning, a question which was answered in the positive. Our findings were visually corroborated both by visualisations using dimensionality reduction techniques showing cleaner clusters in the learned embedding space, and hence less adjacent as well as non-adjacent room confusion, and also by visualisation of pairwise distances showing a clearer contrast between distances to same-class and other-class examples.

Chapter 5. In this chapter, we looked into the performance of a federated learning approach to RSSI-based indoor localisation and compared it to individual learning. We then proposed a local fine-tuning algorithm with acceptance/rejection based on local validation error improvement. This overcame the limitation of prior work where the amount of local fine-tuning, and possibly other hyperparameters, had to be adjusted to the data at hand. We have shown that, given a separation of the model into a global federated and private functions, it is possible to update the private parameters for multiple gradient steps with a high learning rate during the 'self-tuning' and yet retain stable training in terms of variance over samples of the backbone weights sent by the server in the first round. We have shown that in tuned FL each client has a better personalised model than the one they would have had in a benchmark FL evaluation round.

The findings are limited, to an extent, by the datasets, and in Section 6.1 we will discuss amongst other things an ideal dataset, either synthetic or real-world, and methods that, in our opinion, are worth exploring for the realistic continuous learning problem.

6.1 Future Work

A recurring theme in this thesis were representations, or embeddings, of data. However, we did not propose any notion of embedding quality that did not rely on having labeled data. In order to turn visual insights of Chapter 4 into concrete numerical measures of representation quality, one could take inspiration from the UMAP visualisations and make sure that the data in the representation are not too "torn apart" by measuring the degree of connectedness of the graph, for example by computing the number of disconnected components or by looking for bottlenecks. Then, for subsequent classification with kNN, we could take a sufficiently high k value that ensures the graph is connected, and we could rule out some potential representations where the k to ensure connectedness is so large that distinctions between class disappear. An optimisable measure of graph quality in the representation space could also be considered, as for example has been proposed by the authors in [80]. Whether it is at all necessary to learn embeddings for individual datapoints is also a reasonable question. It seems that the answer is no, unless visualisation is needed, as pairwise distances are sufficient for classification using kNN, or the mean linear classifier of equation 2.3, and the fact is that a union of manifolds of different dimensions is likely a better explanation of complex datasets [81].

Because in all cases we were classifying data with temporal structure, if the transition matrix T_y between labels was known, we could assume a linear-chain conditional random field (CRF) model for a sequence of observations of the feature. This allows one to set the likelihood of a sequence of transitions to be strictly zero for any sequence containing a disallowed transition. However, it is quite realistic that we would not have access to a transition matrix, in which case two options can be considered: using pseudo-labels from the classifier labeling a new unlabeled stream to estimate T_y, or, more generally, when representation learning is separated from the classification, estimating the transition model in the representation T_z. As the authors in [82] have shown, learning a binary transition classifier that learns to distinguish between pairs time-consecutive examples and pairs of unrelated samples is guaranteed, with some assumptions, to recover the transition density. If this second route of separating representation learning from classification is taken, then an appropriate CRF model would be the one shown in Figure 6.1.

As for the federated learning exploration of Chapter 5, given the results of individual learning were mostly superior to tuned FL, one might raise the question whether FL is worth it at all. To answer this, a better research question might be: is generalisation better starting from global model backbone, or from random initialisation of the backbone - that is whether a client should ever join FL in the first place. At the first instance the client might ask the question - with the global weights, will the same result, in terms of generalization error, be reached quicker? Indeed in [83] the authors show that a better algorithmic stability, and thus generalisation error, can be achieved with the 'correct' initialisation, one that has low loss on the data.

Finally, the most promising avenue for future exploration, in the author's opinion, would be proper continuous learning, proper in the sense of correctly implementing the ruling out of

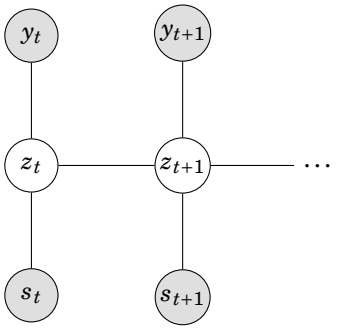


Figure 6.1: A conditional random field model for indoor localisation with a latent variable z. s_t is the observed signal at time t and y_t is the label.

inconsistent hypotheses and thus progressively reducing the hypothesis set size, using consistency regularisation to incorporate the unlabeled data. One such method for continuous learning of neural networks has recently been proposed by the authors in [84], and it would therefore be interesting to combine this with consistency regularisation to investigate what guarantees can be made on target set performance. This should be done concurrently with the collection, or simulation, of an 'ideal' indoor sensing dataset. Firstly, we could have a greater number of clients for experiments in federated learning, and, more importantly, it would consist of an initial fingerprinting, followed by a stream of unlabeled data over a long enough duration such that storage of all data would no longer be feasible, with more labelled collected periodically for experimental verification of the proposed methods, possibly by employing active learning techiques.

BIBLIOGRAPHY

[1] Mohammud J. Bocus, Wenda Li, Jonas Paulavicius, Ryan McConville, Raul Santos-Rodriguez, Kevin Chetty, and Robert Piechocki.
Translation resilient opportunistic wifi sensing.
In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 5627–5633, 2021.

[2] James Pope, Ryan McConville, Michał Kozłowski, Xenofon Fafoutis, Raúl Santos-Rodríguez, R. Piechocki, and I. Craddock.
Sphere in a box: Practical and scalable eurvalve activity monitoring smart home kit.
2017 IEEE 42nd Conference on Local Computer Networks Workshops (LCN Workshops), pages 128–135, 2017.

[3] Michael Holmes, Miquel Perello Nieto, Hao Song, Emma Tonkin, Sabrina Grant, and Peter Flach.
Modelling patient behaviour using IoT sensor data: a case study to evaluate techniques for modelling domestic behaviour in recovery from total hip replacement surgery.
Journal of Healthcare Informatics Research, 4(3):238–260, May 2020.

[4] Rafael Poyiadzi, Weisong Yang, Y. Ben-Shlomo, I. Craddock, L. Coulthard, Raúl Santos-Rodríguez, J. Selwood, and Niall Twomey.
Detecting signatures of early-stage dementia with behavioural models derived from sensor data.
In *AAI4H@ECAI*, 2020.

[5] Mingmin Zhao, Yingcheng Liu, Aniruddh Raghu, Hang Zhao, Tianhong Li, Antonio Torralba, and Dina Katabi.
Through-wall human mesh recovery using radio signals.
In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10112–10121, 2019.

[6] Vladimir Bellavista-Parent, Joaquín Torres-Sospedra, and Antoni Pérez-Navarro.
New trends in indoor positioning based on wifi and machine learning: A systematic review.
In *International Conference on Indoor Positioning and Indoor Navigation, IPIN 2021, Lloret de Mar, Spain, November 29 - Dec. 2, 2021*, pages 1–8. IEEE, 2021.

[7] Yanzi Zhu, Zhujun Xiao, Yuxin Chen, Zhijing Li, Max Liu, Ben Y. Zhao, and Haitao Zheng.
Et tu alexa? when commodity wifi devices turn into adversarial motion sensors.
Proceedings 2020 Network and Distributed System Security Symposium, 2018.

[8] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala.
PyTorch: An Imperative Style, High-Performance Deep Learning Library.
Curran Associates Inc., Red Hook, NY, USA, 2019.

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay.
Scikit-learn: Machine learning in Python.
Journal of Machine Learning Research, 12:2825–2830, 2011.

[10] Eli Bingham, Jonathan P. Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul Szerlip, Paul Horsfall, and Noah D. Goodman.
Pyro: Deep universal probabilistic programming.
Journal of Machine Learning Research, 20(28):1–6, 2019.

[11] Daniel J. Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Javier Fernandez-Marques, Yan Gao, Lorenzo Sani, Kwing Hei Li, Titouan Parcollet, Pedro Porto Buarque de Gusmão, and Nicholas D. Lane.
Flower: A friendly federated learning research framework, 2020.
`https://doi.org/10.48550/arXiv.2007.14390`.

[12] Negar Ghourchian, Michel Allegue-Martinez, and Doina Precup.
Real-time indoor localization in smart homes using semi-supervised learning.
In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 4670–4677. AAAI Press, 2017.

[13] Jonas Paulavičius, Seifallah Jardak, Ryan McConville, Robert Piechocki, and Raul Santos-Rodriguez.
Temporal self-supervised learning for rssi-based indoor localization.
In *ICC 2022 - IEEE International Conference on Communications*, pages 3046–3051, 2022.

[14] Wenda Li, Robert J. Piechocki, Karl Woodbridge, Chong Tang, and Kevin Chetty.
Passive wifi radar for human sensing using a stand-alone access point.
IEEE Transactions on Geoscience and Remote Sensing, 59(3):1986–1998, 2021.

[15] Dallan Byrne, Michal Kozlowski, Raul Santos-Rodriguez, Robert Piechocki, and Ian Craddock.
Residential wearable RSSI and accelerometer measurements with detailed location annotations.
Scientific Data, 5(1), August 2018.

[16] Tianle Cai, Ruiqi Gao, Jason Lee, and Qi Lei.
A theory of label propagation for subpopulation shift.
In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 1170–1182. PMLR, 18–24 Jul 2021.

[17] Maria-Florina Balcan and Avrim Blum.
A discriminative model for semi-supervised learning.
J. ACM, 57(3), mar 2010.

[18] T. Cover and P. Hart.
Nearest neighbor pattern classification.
IEEE Transactions on Information Theory, 13(1):21–27, 1967.

[19] Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim.
On the surprising behavior of distance metrics in high dimensional space.
In Jan Van den Bussche and Victor Vianu, editors, *Database Theory — ICDT 2001*, pages 420–434, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[20] V. Vapnik.
Principles of risk minimization for learning theory.
In J. Moody, S. Hanson, and R.P. Lippmann, editors, *Advances in Neural Information Processing Systems*, volume 4. Morgan-Kaufmann, 1991.

[21] Andrew McRae, Justin Romberg, and Mark Davenport.
Sample complexity and effective dimension for regression on manifolds.
In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12993–13004. Curran Associates, Inc., 2020.

[22] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei.
ImageNet Large Scale Visual Recognition Challenge.
International Journal of Computer Vision (IJCV), 115(3):211–252, 2015.

[23] Phil Pope, Chen Zhu, Ahmed Abdelkader, Micah Goldblum, and Tom Goldstein.
The intrinsic dimension of images and its impact on learning.
In *International Conference on Learning Representations*, 2021.

[24] Ella Bingham and Heikki Mannila.
Random projection in dimensionality reduction: Applications to image and text data.
In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '01, page 245–250, New York, NY, USA, 2001. Association for Computing Machinery.

[25] Ian T. Jolliffe and Jorge Cadima.
Principal component analysis: a review and recent developments.
Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 374(2065):20150202, April 2016.

[26] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford.
A global geometric framework for nonlinear dimensionality reduction.
Science, 290(5500):2319–2323, 2000.

[27] David L. Donoho and Carrie Grimes.
When does geodesic distance recover the true hidden parametrization of families of articulated images?
In *The European Symposium on Artificial Neural Networks*, 2002.

[28] David L. Donoho and Carrie Grimes.
Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data.
Proceedings of the National Academy of Sciences, 100(10):5591–5596, 2003.

[29] Lawrence K. Saul and Sam T. Roweis.
Think globally, fit locally: Unsupervised learning of low dimensional manifolds.
J. Mach. Learn. Res., 4(null):119–155, dec 2003.

[30] Jonathan Crabbé and Mihaela van der Schaar.
Label-free explainability for unsupervised models.
In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 4391–4420. PMLR, 17–23 Jul 2022.

[31] Laurens van der Maaten and Geoffrey Hinton.
Visualizing data using t-sne.
Journal of Machine Learning Research, 9(86):2579–2605, 2008.

[32] Leland McInnes, John Healy, and James Melville.
Umap: Uniform manifold approximation and projection for dimension reduction, 2018.

[33] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal.
Reconciling modern machine-learning practice and the classical bias–variance trade-off.
Proceedings of the National Academy of Sciences, 116(32):15849–15854, 2019.

[34] Kurt Hornik, Maxwell Stinchcombe, and Halbert White.
Multilayer feedforward networks are universal approximators.
Neural Networks, 2(5):359–366, 1989.

[35] Zeke Xie, Issei Sato, and Masashi Sugiyama.
A diffusion theory for deep learning dynamics: Stochastic gradient descent exponentially favors flat minima.
In *International Conference on Learning Representations*, 2021.

[36] Lewis Smith and Yarin Gal.
Understanding measures of uncertainty for adversarial example detection.
In Amir Globerson and Ricardo Silva, editors, *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018, Monterey, California, USA, August 6-10, 2018*, pages 560–569. AUAI Press, 2018.

[37] Elad Hoffer, Itay Hubara, and Daniel Soudry.
Train longer, generalize better: Closing the generalization gap in large batch training of neural networks.
In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 1729–1739, Red Hook, NY, USA, 2017. Curran Associates Inc.

[38] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He.
Accurate, large minibatch sgd: Training imagenet in 1 hour, 2017.

[39] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang.
On large-batch training for deep learning: Generalization gap and sharp minima.
In *International Conference on Learning Representations*, 2017.

[40] Ben London.
A pac-bayesian analysis of randomized learning with application to stochastic gradient descent.
In *Neural Information Processing Systems*, 2017.

[41] Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, R. Howard, Wayne Hubbard, and Lawrence Jackel.
Handwritten digit recognition with a back-propagation network.
In D. Touretzky, editor, *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann, 1989.

[42] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams.
Learning internal representations by error propagation.
1986.

[43] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio.
On the properties of neural machine translation: Encoder–decoder approaches.
In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar, October 2014. Association for Computational Linguistics.

[44] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun.
Deep residual learning for image recognition.
In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[45] P.L. Bartlett.
The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network.
IEEE Transactions on Information Theory, 44(2):525–536, 1998.

[46] Amit Daniely.
Depth separation for neural networks.
In Satyen Kale and Ohad Shamir, editors, *Proceedings of the 2017 Conference on Learning Theory*, volume 65 of *Proceedings of Machine Learning Research*, pages 690–696. PMLR, 07–10 Jul 2017.

[47] Gregory Naitzat, Andrey Zhitnikov, and Lek-Heng Lim.
Topology of deep neural networks.
J. Mach. Learn. Res., 21(1), jan 2020.

[48] Hongzhou Lin and Stefanie Jegelka.
Resnet with one-neuron hidden layers is a universal approximator.
In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[49] Boris Hanin and David Rolnick.
Complexity of linear regions in deep networks.
In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2596–2604. PMLR, 09–15 Jun 2019.

[50] R. Hadsell, S. Chopra, and Y. LeCun.
Dimensionality reduction by learning an invariant mapping.
In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742, 2006.

[51] Tongzhou Wang and Phillip Isola.
Understanding contrastive representation learning through alignment and uniformity on the hypersphere.
In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9929–9939. PMLR, 13–18 Jul 2020.

[52] Feng Wang and Huaping Liu.
Understanding the behaviour of contrastive loss.
In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2495–2504, 2021.

[53] Roland S. Zimmermann, Yash Sharma, Steffen Schneider, Matthias Bethge, and Wieland Brendel.
Contrastive learning inverts the data generating process.
In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12979–12990. PMLR, 18–24 Jul 2021.

[54] Han Bao, Yoshihiro Nagano, and Kento Nozawa.
On the surrogate gap between contrastive and supervised losses.
In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 1585–1606. PMLR, 17–23 Jul 2022.

[55] Nikunj Saunshi, Orestis Plevrakis, Sanjeev Arora, Mikhail Khodak, and Hrishikesh Khandeparkar.
A theoretical analysis of contrastive unsupervised representation learning.

In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5628–5637. PMLR, 09–15 Jun 2019.

[56] Marta Horvath and Gábor Lugosi.
A data-dependent skeleton estimate and a scale-sensitive dimension for classification.
Economics Working Papers 199, Department of Economics and Business, Universitat Pompeu Fabra, December 1996.

[57] Jeff Z. HaoChen, Colin Wei, Adrien Gaidon, and Tengyu Ma.
Provable guarantees for self-supervised deep learning with spectral contrastive loss.
In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[58] Jeff Z. HaoChen, Colin Wei, Ananya Kumar, and Tengyu Ma.
Beyond separability: Analyzing the linear transferability of contrastive representations to related subpopulations.
In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[59] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas.
Communication-Efficient Learning of Deep Networks from Decentralized Data.
In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR, 20–22 Apr 2017.

[60] Tao Yu, Eugene Bagdasaryan, and Vitaly Shmatikov.
Salvaging federated learning by local adaptation, 2020.
`https://doi.org/10.48550/arXiv.2002.04758`.

[61] J. Capon.
High-resolution frequency-wavenumber spectrum analysis.
Proceedings of the IEEE, 57(8):1408–1418, 1969.

[62] Diederik P Kingma and Max Welling.
Auto-encoding variational bayes, 2013.

[63] S. M. Ali Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, koray kavukcuoglu, and Geoffrey E Hinton.
Attend, infer, repeat: Fast scene understanding with generative models.
In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[64] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and koray kavukcuoglu.
Spatial transformer networks.
In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[65] Enrique García, Pablo Poudereux, Álvaro Hernández, Jesús Ureña, and David Gualda.
A robust uwb indoor positioning system for highly complex environments.
In *2015 IEEE International Conference on Industrial Technology (ICIT)*, pages 3386–3391, 2015.

[66] Wenda Li, Bo Tan, and Robert Piechocki.
Opportunistic doppler-only indoor localization via passive radar.
In *2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*, pages 467–473, 2018.

[67] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan.
Supervised contrastive learning.
In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 18661–18673. Curran Associates, Inc., 2020.

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

[69] Shuo Chen, Gang Niu, Chen Gong, Jun Li, Jian Yang, and Masashi Sugiyama.
Large-margin contrastive learning with distance polarization regularizer.
In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 1673–1683. PMLR, 18–24 Jul 2021.

[70] Xin Cheng, Chuan Ma, Jun Li, Haiwei Song, Feng Shu, and Jiangzhou Wang.
Federated learning-based localization with heterogeneous fingerprint database.
IEEE Wireless Communications Letters, pages 1–1, 2022.

[71] Zheshun Wu, Xiaoping Wu, and Yunliang Long.
Prediction based semi-supervised online personalized federated learning for indoor localization.
IEEE Sensors Journal, pages 1–1, 2022.

[72] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar.
Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach.
In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[73] Peng Wu, Tales Imbiriba, Junha Park, Sunwoo Kim, and Pau Closas.
Personalized federated learning over non-iid data for indoor localization.
In *2021 IEEE 22nd International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pages 421–425, 2021.

[74] Yutao Huang, Lingyang Chu, Zirui Zhou, Lanjun Wang, Jiangchuan Liu, Jian Pei, and Yong Zhang.
Personalized cross-silo federated learning on non-iid data.
Proceedings of the AAAI Conference on Artificial Intelligence, 35(9), 2021.

[75] Junha Park, Jiseon Moon, Taekyoon Kim, Peng Wu, Tales Imbiriba, Pau Closas, and Sunwoo Kim.
Federated learning for indoor localization via model reliability with dropout.
IEEE Communications Letters, pages 1–1, 2022.

[76] Xin Cheng, Chuan Ma, Jun Li, Haiwei Song, Feng Shu, and Jiangzhou Wang.
Federated learning-based localization with heterogeneous fingerprint database.
IEEE Wireless Communications Letters, 11(7), 2022.

[77] Ilya Loshchilov and Frank Hutter.
Decoupled weight decay regularization.
In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[78] Ashia C. Wilson, Rebecca Roelofs, Mitchell Stern, Nathan Srebro, and Benjamin Recht.
The marginal value of adaptive gradient methods in machine learning.
In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017.

[79] Sashank J. Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and Hugh Brendan McMahan.
Adaptive federated optimization.
In *International Conference on Learning Representations, ICLR*, 2021.

[80] Konstantinos Kamnitsas, Daniel Castro, Loic Le Folgoc, Ian Walker, Ryutaro Tanno, Daniel Rueckert, Ben Glocker, Antonio Criminisi, and Aditya Nori.

Semi-supervised learning via compact latent space clustering.
In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2459–2468. PMLR, 10–15 Jul 2018.

[81] Bradley CA Brown, Anthony L. Caterini, Brendan Leigh Ross, Jesse C Cresswell, and Gabriel Loaiza-Ganem.
The union of manifolds hypothesis.
In *NeurIPS 2022 Workshop on Symmetry and Geometry in Neural Representations*, 2022.

[82] Bingbin Liu, Pradeep Ravikumar, and Andrej Risteski.
Contrastive learning of strong-mixing continuous-time stochastic processes.
In *International Conference on Artificial Intelligence and Statistics*, 2021.

[83] Ilja Kuzborskij and Christoph Lampert.
Data-dependent stability of stochastic gradient descent.
In *Proceedings of the 35th International Conference on Machine Learning*, volume 80. PMLR, 2018.

[84] Maciej Wołczyk, Karol Piczak, Bartosz Wójcik, Lukasz Pustelnik, Paweł Morawiecki, Jacek Tabor, Tomasz Trzcinski, and Przemysław Spurek.
Continual learning with guarantees via weight interval constraints.
In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 23897–23911. PMLR, 17–23 Jul 2022.