# Temporal Self-Supervised Learning for RSSI-based Indoor Localization

Jonas Paulavičius
*University of Bristol*
*Toshiba Research Europe*

Seifallah Jardak
*Toshiba Research Europe*

Ryan McConville
*University of Bristol*

Robert Piechocki
*University of Bristol*

Raul Santos-Rodriguez
*University of Bristol*

ICC 2022 - IEEE International Conference on Communications | 978-1-5386-8347-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICC45855.2022.9838841

***Abstract*—The feasibility of integrating the temporal nature of the Bluetooth Low Energy (BLE) Received Signal Strength Indicator (RSSI) into a self-supervised machine learning model for room-level and sub-room-level localization in a realistic residential setting is investigated. The signal is transmitted by a wearable wrist watch and received by multiple access points acting as receivers communicating via the BLE standard. It is found that while the baseline room-level accuracy is sufficiently high for practical applications in rooms separated by a wall, confusion can occur between non-adjacent rooms and thus lower localization performance. Two approaches are explored that exploit the time dimension of the data to mitigate this problem: maximum likelihood estimation in a conditional random field model, and self-supervised contrastive learning based on temporal proximity. Using a real world dataset collected in residential homes, we develop the approaches on the data collected in one residence before evaluating them on data collected in another. On the evaluation residence, we find that conditional random fields do not improve upon the baseline in terms of the weighted F1 score, while contrastive learning leads to an improvement in localization performance.**

***Index Terms*—localization, RSSI, fingerprinting, contrastive learning, e-health**

## I. Introduction

Indoor localization based on pervasive radio frequency (RF) technologies has numerous applications, particularly related to healthcare, including early diagnosis of cognitive disorders such as dementia [1] and assisting people with visual impairments inside buildings [2].

Localization methods fall into two broad categories - direct methods and fingerprinting. Direct methods are typically limited, for example to line of sight scenarios in the case of Ultra-wide band triangulation [3], or require extra information such as initial position, for example in dead-reckoning with bistatic WiFi radar [4]. In this work we focus on fingerprinting, which is a method where a signal map of the environment is created prior to localization. It is most promising as it requires neither light nor line of sight to work.

We investigate fingerprinting using the Received Signal Strength Indicator (RSSI) in the Bluetooth Low Energy (BLE) protocol. While it is inexpensive and easy to collect, the difficulty with using RSSI is that 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 demonstrate how this issue manifests itself in a residential setting. Specifically, we show that confusion can occur between rooms that are not necessarily adjacent which results in localization errors. While this can be mitigated, to some extent, by adding extra Access Points (AP), it is not always practical or cost efficient to do so. We compare two methods of mitigating the high variability of RSSI after the deployment of the system by exploiting the time dimension:

1) Maximum likelihood estimation (MLE) in a linear-chain conditional random field (CRF) model. This is possible when the house layout and possible transitions between a set of pre-defined tiles are known.
2) Contrastive learning (CL), a self-supervised method requiring only unlabeled data to learn a nonlinear transformation of RSSI to an embedding space where the data must obey specified constraints in this space.

We will evaluate each of the above approaches on a real world dataset of BLE RSSI data collected from residential houses [5]. We will show that MLE with the CRF does not improve localization performance over baselines, with an F1 score of 0.82 compared to the 0.83 baseline score, even with apriori knowledge of the environment. However, the self-supervised contrastive learning approach, which specifically accounts for the temporal nature of the data, does improve the localization performance with an F1 score of 0.86, without the need for any apriori knowledge of the environment.

The rest of the paper is organised as follows. Section II gives an overview of relevant work in applications of metric learning to RF fingerprint-based indoor localisation. Section III introduces the dataset. Section IV details the methods used. In Section V, the results are presented and in Section VI the implications of these results are discussed. Section VII concludes this paper.

## II. Related work

The feasibility of room-level localization is confirmed by meta-review of indoor positioning [6] which found low median errors (2m-5m) when using BLE fingerprinting. In the limit of a high density of APs, specifically 24 BLE beacons on the walls of a 16m by 2m corridor, fingerprinting can achieve under 1m error at 83 % of the cumulative distribution function (CDF) [7].

The authors in [8] investigate using RSSI between BLE beacons and various smartphones for indoor localization, find-

ing high variance of results between smartphones. To mitigate that, a shared embedding is learned based on the constraint that the relationship between RSSI values for nearby beacons should be independent of the smartphone. For phones for which labeled data were not explicitly collected, they find the greatest improvement reduces the mean absolute error from 2.62 meters to 1.63 meters.

Another cause of variation is environment changes in time. In [9] the authors optimize the triplet margin loss with a neural network embedding to make "outdated" Channel State Information (CSI) signals close to"fresh" CSI on a few of the APs. An average of 0.2m error reduction is obtained without requiring full re-fingerprinting.

A similar setting, where the environment change is enacted by moving one or two APs, is studied in [10], also for the case of CSI fingerprinting. To mitigate the environment change, the approach uses transfer learning to find a shared embedding for the fingerprints before and after the change. The work reports a 2m error at 90 % CDF, rising by only $\sim$ 0.2 m if an AP is moved, without the need for re-fingerprinting.

In [11], the authors propose to learn a Mahalanobis metric that better respects notion of distance similarity. The test is carried out in a large indoor area, finding an improvement from 2.3m of baseline kNN to 2m at 90% CDF. However, this method uses labeled data for metric learning.

This present work proposes a an effective, yet less complex approach, applicable in all cases where streaming signal data is available, by imposing a simple constraint on the embedding - that it should be invariant to factors other than the position of the transmitter, which we demonstrate empirically to improve performance.

## III. Dataset

We use the dataset collected by [5] and follow the same designation as in the paper. This dataset contains several hours of recordings of time-aligned BLE RSSI and position annotations at roughly meter squared precision in several residential houses.

Communication with each AP happens roughly every 200 ms. The position annotations are available roughly every 40 ms. The tag ID, representing the location, of a 0.2 second period is chosen as the most frequent tag in that period. The default RSSI value for when an AP is has not communicated with the wearable with is set to -108dB. All RSSI data are normalized to be in [-1, 1] using the pretraining dataset to estimate amplitude.

Of the data available, houses C and D are chosen for all experiments, as these have the greatest volume of data collected. It is ensured that all tiles are visited in the training, validation and testing datasets. It is also ensured there are no disallowed transitions in the data. House D is used for development of the proposed models, while House C is used for evaluation of the proposed models. These will be referred to as the development house and the evaluation house respectively in what follows.

The 8 room-level tags for the development house with their IDs are given in Table I. The allowed room-level transitions are: 0-1, 0-2, 2-3, 1-5, 1-4, 4-5, 5-6, 5-7.

| | | | |
|---|---|---|---|
| 0. hallway_lower | 1. stairs | 2. living_area | 3. kitchen |
| 4. bathroom_toilet | 5. hallway_upper | 6. bedroom_1 | 7. bedroom_2 |

TABLE I: Room-level tags and their IDs for the development house.

The sub-room-level tags for the development house with their IDs are given in Table II. Metre-precision tags are given in brackets next to the sub-room-level tag name where it is not clear from the naming which metre-precision tags might fall under the given sub-room-level tag. The allowed sub-room-level transitions are: 0-1, 0-2, 2-3, 2-4, 2-6, 3-4, 3-5, 1-8, 1-7, 7-8, 8-9, 9-10, 8-11.

| | |
|---|---|
| 0. hallway_lower | 1. stairs |
| 2. living_area_A1 (4, 9, 10, 15) | 3. living_area_A2 (5, 6, 7, 8) |
| 4. living_area_A3 (11, 12, 13, 14) | 5. living_area_B |
| 6. kitchen | 7. bathroom_toilet |
| 8. hallway_upper | 9. bedroom_1_1 (30, 31, 32, 33) |
| 10. bedroom_1_2 (34, 35, 36, 37) | 11. bedroom_2 |

TABLE II: Sub-room-level tags (metre-precision tag in brackets where unclear from naming) and their IDs for the development house.

The data splits for experiments on the development house are the following: 'Fingerprint rapid' (4 min) or 'Fingerprint floor' (1h) are used for training, 'Living 5' (43 min) is used for validation and 'Living 2' (59 min) is used for self-supervised learning.

In contrast to 'Fingerprint rapid' where each tile is visited, 'Fingerprint floor' contains additional samples where the data collector stands facing each of the four directions on every tile. Although having a labeled "living" sequence can be useful for validation and sequence prediction, this sequence might not be available in a realistic scenario given the arduous nature of the collection. For that reason, realistic data splits are used for the evaluation house.

The 8 room-level tags for the evaluation house with their IDs are given in Table III. The allowed room-level transitions are: 0-1, 0-2, 1-8, 2-3, 3-4, 3-5, 3-6, 3-7, 6-7.

| | | |
|---|---|---|
| 0. living_room | 1. kitchen | 2. stairs |
| 3. hallway_upper | 4. study | 5. bedroom_1 |
| 6. bathroom_toilet | 7. bedroom_2 | 8. outside |

TABLE III: Room-level tags and their IDs for the evaluation house.

We use the following data splits for experiments on the evaluation house: 80 % of 'Fingerprint rapid' (5 min) is used for training, 20 % of 'Fingerprint rapid' (1 min) for validation, 'Living 1', 'Living 6' and 'Living 10' (2h 24min) for self-supervised learning and 'Living 2', 'Living 3', 'Living 4', 'Living 5', 'Living 7', 'Living 8' and 'Living 9' (1h 33 min) for testing.

## IV. METHODOLOGY

### A. Supervised Learning Baseline

To set a baseline, we compare three algorithms: a k-Nearest-Neighbour (kNN) classifier implemented in scikit-learn [12], logistic regression with the default parameters as implemented in scikit-learn, and neural networks (NN) implemented using the Pytorch [13] library.

The metric used to choose hyperparameters is the F1 score weighted by the true label proportion on the validation set. For kNN, $k \in \{1, 2, ..., 40\}$ and distance-dependent weights are used for prediction as this outperforms uniform weights. $L_1$ distance is used in the feature space as it is found to be superior to $L_2$ distance.

It is first assessed how the performance changes with using both a single timestep and longer RSSI sequences as the feature. It is found that little improvement is made beyond using 5 timesteps, thus the $(N_{AP} \times 5)$-dimensional vector $R$ is used as the feature from now on unless stated otherwise.

The NN tested are: a feedforward NN (FNN) and a Gated Recurrent Unit (GRU, [14]) with default parameters, where the feature for the latter is (5, $N_{AP}$)-dimensional. For the FNN, we use ReLu activations and residual linear connections between each consecutive layer. We vary the number of layers L between 1 and 3. For L = 1, the hidden layer size N is varied between 50 and 500. For L = 2, N = 10, 20, ..., 50 are evaluated. For L = 3, various layer widths between N = 10 and N = 50 are considered.

The loss function used for training the NN is the categorical cross-entropy weighted by the inverse class distribution and mixup [15]. Stochastic gradient descent with the AdamW [16] optimizer is used with the default hyperparameters: a learning rate of $10^{-3}$ and a weight decay factor of $10^{-2}$. Each gradient step is taken over the full training dataset of size 605.

### B. Maximum Likelihood Sequence Prediction

We assume a linear-chain conditional random field (CRF) model [17] for a sequence of observations of the feature $x_{1:T} = [x_1...x_T]$, where the square brackets denote a tuple. The marginal likelihood of the sequence of positions $y_{1:T} = [y_1...y_T]$ is

$$p(y_{1:T}|x_{1:T}) = \frac{1}{Z(x_{1:T})} F(x_{1:T}, y_{1:T}) \tag{1}$$

where $Z$ is the normalisation constant and $F(x_{1:T}, y_{1:T}) = f(x_1, y_1)T(x_1, x_2)f(x_2, y_2)..T(x_{T-1}, x_T)f(x_T, y_T)$ for a linear-chain CRF, with $f(x_t, y_t)$ the discrete distribution over labels output by the classifier and $T$ the transition matrix. This allows one to set the likelihood of a sequence of transitions to be strictly zero for any sequence containing a disallowed transition..

The Viterbi maximum likelihood algorithm [18] is used to estimate the sequence $x$ that maximizes the marginal likelihood. Logarithms of all quantities are used in order to convert products to sums to avoid arithmetic underflow.

With no prior information, the transition matrix is simply a matrix with entries equal to 1 for allowed transitions, including staying on same tile (same location), and 0 for disallowed transitions. It is normalized so entries are probabilities for going i to j. Increasing the diagonal entries to be equal to 3, or 5 in the case of the room-level transition matrix, before normalization is found to improved performance. This is reasonable, as we expect the person to more likely remain in the same tile rather than change tiles at any given time.

### C. Contrastive Learning

The goal is to find a better representation/embedding $z = f(x)$ of the feature $x$. We want this representation to have the property that for a small change in the person's position $y$, the change in $z$ is also small. This should therefore lead to the representation becoming insensitive to factors that we are not interested in, such as orientation or pose. This can be achieved by minimizing the classification softmax loss [19]:

$$L(\theta) = \mathop{\mathbb{E}}_{\substack{x\sim p \\ x^+\sim p_x^+ \\ \{x_k^-\}_{k=1..K}\sim p}} \left[ -\ln \frac{\mathrm{e}^{-\mathrm{d(z,z^+)}}}{\mathrm{e}^{-\mathrm{d(z,z^+)}} + \sum_{\mathrm{k}} \mathrm{e}^{-\mathrm{d(z,z_k^-)}}} \right] \tag{2}$$

where $z = f_\theta(x)$ with $f_\theta$ the embedding function family parameterized by $\theta$ and $d$ the distance function in the embedding space. $x$ is called the anchor, $p$ is the data distribution, and we estimate the expectation by minibatches from the pretraining dataset. $x^+$ is the positive example, $p^+$ is the distribution of positive examples, over which we estimate the expectation using as the single positive example the RSSI feature 5 steps (1 second) into the future. The $K$ samples $\{x_k^-\}$ are called the negative examples.

As we have some labeled data available, and a classifier trained using this data, it is possible to predict room labels for the pretraining set. We do so at the start of training for the whole pretraining dataset. Using this information, we use instead of $L$ the loss function $L_{debiased}$ as given in eqn. 8 of [20], with $N = K$ and $M = 1$:

$$L_{\text{debiased}}^{N,M} = \mathop{\mathbb{E}}_{\substack{x\sim p;\, x^+\sim p_x^+ \\ \{u_i\}_{i=1}^N\sim p \\ \{v_i\}_{i=1}^N\sim p_x^+}} \left[ -\ln \frac{e^{-d(z,z^+)}}{e^{-d(z,z^+)} + Ng\Big(x, \{u_i\}, \{v_i\}\Big)} \right] \tag{3}$$

with

$$g(x, \{u_i\}_{i=1}^N, \{v_i\}_{i=1}^M) = \max\Big\{ \frac{1}{\tau^-}\Big( \frac{1}{N}\sum_{i=1}^N e^{-d(f(x),f(u_i))} - \tau^+ \frac{1}{M}\sum_{i=1}^M e^{-d(f(x),f(v_i))} \Big),\ e^{-l} \Big\}, \tag{4}$$

$\tau^+$ being the proportion of the dataset with the same class as that of $x$ , $\tau^- = 1 - \tau^+$ and $l$ the maximum possible value of $d(z, z')$ for any $z, z'$.

The common choice of embedding normalized to the unit hypersphere is considered. The distance function used is the cosine distance, $d(z, z') = 1 - z \cdot z'$. Minimizing this loss function minimizes the distances between positive examples and,

in the limit of infinite negative examples, enforces uniformity on the hypersphere [21].

For $f_\theta$, the GRU is used. The embedding dimension is 5. That the representation should be lower dimensional than the input is desirable as the distance distribution between points in high dimensions becomes increasingly narrowly peaked (see Section 4.5 of [22] for the specific case of uniform distribution on a n-dimensional hypersphere). The size of the hidden dimension of the GRU is set to 100 for the development house and 150 for the evaluation house.

To minimize the loss we use gradient descent with the AdamW optimizer with the default hyperparameter settings. We fix the batch size at 512 and vary the number of negative examples.

## V. Results

### A. Baseline

In the development house, for room-level classification using kNN, the best $k$ value is found to be 7, with an F1 score of 0.89. Using 'Fingerprint floor' as the training set, the best $k$ value is found to be 27, with an F1 score of 0.92. For sub-room-level classification, the best $k$ value is found to be 17, with an F1 score of 0.66. If 'Fingerprint floor' is used as the training set, the best $k$ value is found to be 20 with an F1 score of 0.70. We do not evaluate any further the sub-room-level classification results for the sub-room-level tiles defined in Section III as these are found to be not good enough for practical application.

Logistic regression for room-level classification converges to a solution with a training set F1 score of 0.971. On the validation set, the F1 score is 0.762, which is significantly lower than the kNN. Of the NN and loss functions evaluated, GRU with a hidden dimension size of 50 trained with mixup performs best with a room-level weighted F1 score of 0.91, which again is lower than the performance of the kNN.

For the evaluation house, the best $k$ value is found to be 3 (scoring 0.88 on the validation dataset). The weighted F1 score is 0.83 for room-level classification. The confusion matrix is shown in Figure 1.

### B. Maximum Likelihood Sequence Prediction

For room-level classification in the development house, the best $k$ is 8 with an F1 score of 0.90.

For the evaluation house, using $k$ = 3, the weighted F1 score is found to be 0.82 for room-level classification, which is small drop in performance, relative to the baseline. The confusion matrix is shown in Figure 2.

### C. Contrastive Learning

We assess the results for the development house with $K = 16$ and $K = 256$, with the result given as the mean $\pm$ standard deviation of 5 training runs. The random number generator (RNG) for minibatch sampling and the RNG for negative sampling are given the same seed, but the neural network weights at initialization are sampled with a different


Fig. 1: Room level confusion matrix for the evaluation house using the $L_1$ weighted k(=3)NN. Note the confusion between non-adjacent rooms such as living_room and bathroom_toilet and living_room and bedroom_1.

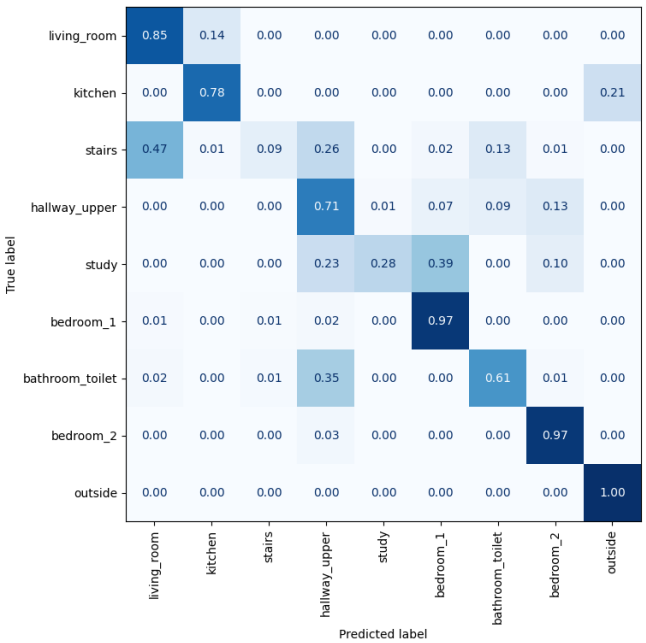


Fig. 2: Room-level confusion matrix for the evaluation house using $L_1$ weighted k(=3)NN and Viterbi max likelihood. While there is less confusion between non-adjacent rooms as compared to the baseline (Figure 1), the confusion between living room and bedroom 1 is anomalous here given that they are separated by more than one transition.

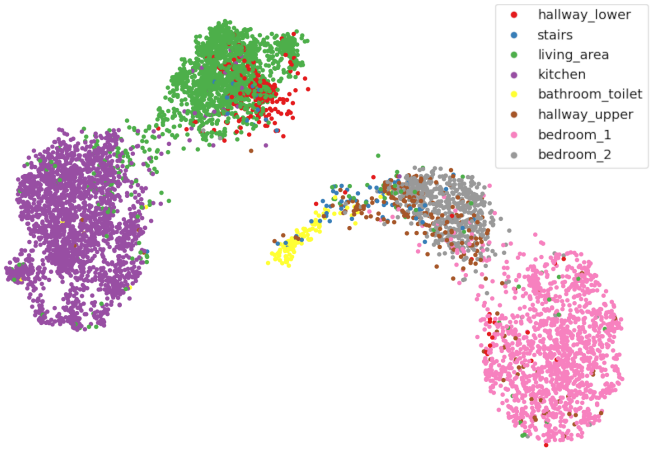


Fig. 3: UMAP (k = 10, min dist = 0.1) visualization of the raw pretraining dataset. The ground truth room-level labels are displayed.

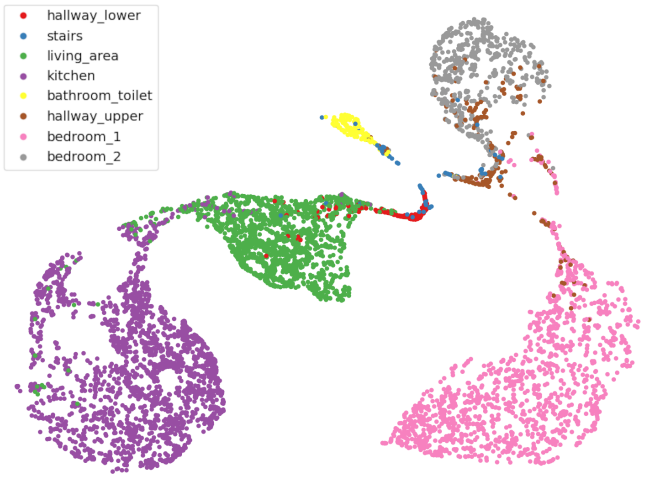


Fig. 4: UMAP ($k$ = 10, min dist = 0.1) visualization of the pretraining dataset after the embedding learned by optimising $L_{debiased}$ with $K = 16$. The ground truth room-level labels are displayed.

seed. For each training run of a given hyperparameter configuration, when evaluating, the kNN (in the embedding space) result which scores highest for $k \in \{1, 2, ..., 40\}$ is used in computing the statistics (mean, standard deviation) given.

$K = 256$ scores 0.91 $\pm$ 0.01 while $K = 16$ slightly improves this, giving the result 0.92 $\pm$ 0.00 on the room-level classification task. We fix $K = 16$ from here on. The best-performing initialization is used as the embedding function

The observation that the transformation learned is more useful for classification based on nearest neighbours is supported by the visualization of the distribution of pre-training data projected to 2-dimensions before (Figure 3) and after the embedding (Figure 4) using UMAP [23].

Using logistic regression on the embedding space learned by minimizing $L_{debiased}$ with $K = 16$, a room-level **training set** F1 score of 0.85 $\pm$ 0.01 is obtained. On the validation set, the F1 score is 0.91 $\pm$ 0.01. A feedforward NN with L = 2 layers of width (100, 100) is found to perform best in the embedding. The model during training which scored highest in terms of validation F1 score achieves a room-level F1 score of 0.91 $\pm$ 0.00.

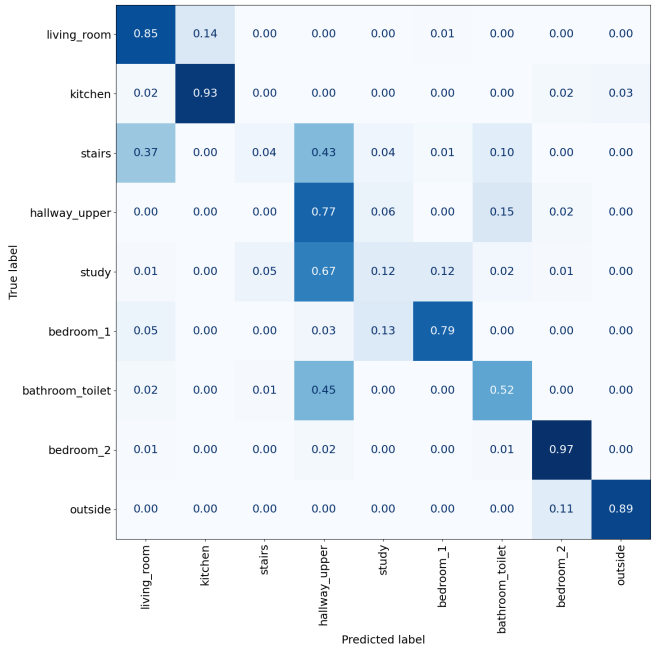


Fig. 5: Room-level confusion matrix for the evaluation house, using $L_2$ weighted k(=6)NN on the embedding space learned by minimizing $L_{debiased}$ with $K = 16$.

As neither logistic regression nor Neural Networks demonstrated an improvement over kNN in the embedding space, we only investigate the result of kNN in the embedding space in the evaluation house. For the evaluation house, a minibatch size of 4096 with $K = 16$ is used with an AdamW learning rate of 0.002 for 250 epochs. An F1 score of 0.86 $\pm$ 0.00 is obtained on the test set. The resulting test set confusion matrix is shown in Figure 5. A summary of the results is given in Table IV.

| | kNN | kNN + Viterbi | kNN + CL |
|---|---|---|---|
| Development house, room | 0.89 | 0.90 | **0.92** |
| Development house, sub-room | 0.66 | **0.72** | 0.61 |
| Evaluation house, room | 0.83 | 0.82 | **0.86** |

TABLE IV: A summary of the results obtained, given as class-weighted F1 scores, of the methods evaluated against the baseline in both houses.

## VI. Discussion

In Section V alternatives for mitigating the high variability of RSSI with application to fingerprint based indoor localization were explored and shown to improve the indoor localization performance in residential settings. While the maximum likelihood inference method in a linear-chain CRF

has benefits, such as less hyperparameter tuning required, it requires the knowledge of the transition matrix. If that is not available, we show that performance is improved when using a contrastive learning method which incorporates temporality on unlabeled data. This method instead relies on the assumption that at nearby timepoints, the location of the person should not vary much.

## VII. Conclusion

In this work we improved on kNN operating on RSSI signals by encoding invariances related to movements of the person that are not changes in their position. To achieve this we proposed and evaluated two models, one of which, a temporal self-supervised based approach, improved the localization performance on a real world dataset collected in residential homes. By learning an embedding of a lower dimension, we also addressed an important issue, specifically that the distance distribution for data in high-dimensional spaces is increasingly peaked.

## VIII. Future Work

While we have demonstrated an improvement in localization performance, several issues remain to be addressed. First of all, the effect of class imbalance in the pretraining dataset was not mitigated in any way. It is clear that performance suffers for rooms that are not the main rooms one might spend time in the house in, such as corridors. In the embedding space, this seems to result in tight clusters for the main rooms and small, pulled apart clusters for the low-data rooms. Secondly, there is the slow convergence of the loss close to the optimum due to the choice of negative examples.

In addition to that, the main question of how to evaluate the "goodness" of a representation without an extensive validation dataset remains. Ideally, one would like that classes are linearly separable at convergence. However, measuring during training the performance of a logistic regression classifier on the training set in fact shows that this linear separability of the embedding does not correlate with kNN accuracy in the embedding.

Finally, it would be useful to add extra constraints, informed by the understanding of the physical properties of the RSSI data, that would enforce some structure on the embedding.

## References


[1] R. Poyiadzi, W. Yang, Y. Ben-Shlomo, I. Craddock, L. Coulthard, R. Santos-Rodríguez, J. Selwood, and N. Twomey, "Detecting signatures of early-stage dementia with behavioural models derived from sensor data," in *AAI4H@ECAI*, 2020.

[2] G. Fusco and J. M. Coughlan, "Indoor localization for visually impaired travelers using computer vision on a smartphone," in *Proceedings of the 17th International Web for All Conference*, W4A '20, (New York, NY, USA), Association for Computing Machinery, 2020.

[3] E. Garca, P. Poudereux, . Hernndez, J. Urea, and D. Gualda, "A robust uwb indoor positioning system for highly complex environments," in *2015 IEEE International Conference on Industrial Technology (ICIT)*, pp. 3386–3391, 2015.

[4] W. Li, B. Tan, and R. 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)*, pp. 467–473, 2018.

[5] D. Byrne, M. Kozlowski, R. Santos-Rodriguez, R. Piechocki, and I. Craddock, "Residential wearable RSSI and accelerometer measurements with detailed location annotations," *Scientific Data*, vol. 5, Aug. 2018.

[6] G. M. Mendoza-Silva, J. Torres-Sospedra, and J. Huerta, "A meta-review of indoor positioning systems," *Sensors*, vol. 19, no. 20, 2019.

[7] D. Ahmetovic, C. Gleason, C. Ruan, K. Kitani, H. Takagi, and C. Asakawa, "Navcog: A navigational cognitive assistant for the blind," in *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '16, (New York, NY, USA), p. 9099, Association for Computing Machinery, 2016.

[8] H. Agarwal, N. Sanghvi, V. Roy, and K. Kitani, "Deepble: Generalizing rssi-based localization across different devices," 2021.

[9] X. Zeng, L. Xiao, M. Zhao, X. Xu, and Y. Li, "Transformable Fingerprinting with Deep Metric Learning Approach for Indoor Localization," in *Journal of Physics Conference Series*, vol. 1575 of *Journal of Physics Conference Series*, p. 012001, June 2020.

[10] Z. Gao, Y. Gao, S. Wang, D. Li, and Y. Xu, "Crisloc: Reconstructable csi fingerprinting for indoor smartphone localization," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3422–3437, 2021.

[11] Y. Zhang, L. Ma, Y. Xu, and Y. Sun, "An rss pathloss considered distance metric learning for fingerprinting indoor localization," in *2019 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, 2019.

[12] 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*, vol. 12, pp. 2825–2830, 2011.

[13] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," 2019.

[14] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," 2014.

[15] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," 2018.

[16] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2019.

[17] C. Sutton and A. McCallum, "An introduction to conditional random fields," 2010.

[18] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.

[19] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," 2020.

[20] C.-Y. Chuang, J. Robinson, L. Yen-Chen, A. Torralba, and S. Jegelka, "Debiased contrastive learning," 2020.

[21] T. Wang and P. Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," 2020.

[22] S. Lellouche and M. Souris, "Distribution of distances between elements in a compact set," *Stats*, vol. 3, no. 1, pp. 1–15, 2020.

[23] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," 2020.