EWOk: Towards Efficient Multidimensional Compression of Indoor Positioning Datasets

Lucie Klus, *Student Member, IEEE,* Roman Klus, *Student Member, IEEE,* Joaquín Torres-Sospedra, Elena Simona Lohan, *Senior Member, IEEE,* Carlos Granell and Jari Nurmi, *Senior Member, IEEE*

Abstract—Indoor positioning performed directly at the end-user device ensures reliability in case the network connection fails but is limited by the size of the RSS radio map necessary to match the measured array to the device's location. Reducing the size of the RSS database enables faster processing, and saves storage space and radio resources necessary for the database transfer, thus cutting implementation and operation costs, and increasing the quality of service. In this work, we propose EWOk, an Element-Wise cOmpression using k-means, which reduces the size of the individual radio measurements within the fingerprinting radio map while sustaining or boosting the dataset's positioning capabilities. We show that the 7-bit representation of measurements is sufficient in positioning scenarios, and reducing the data size further using EWOk results in higher compression and faster data transfer and processing. To eliminate the inherent uncertainty of k-means we propose a data-dependent, non-random initiation scheme to ensure stability and limit variance. We further combine EWOk with principal component analysis to show its applicability in combination with other methods, and to demonstrate the efficiency of the resulting multidimensional compression. We evaluate EWOk on 25 RSS fingerprinting datasets and show that it positively impacts compression efficiency, and positioning performance.

Index Terms—clustering, compression, dimensionality reduction, fingerprinting, indoor positioning, k-means, k-nearest neighbors, on-device computing

1 Introduction

Performing localization and positioning in indoor environments on end-user devices is a crucial requirement for various mobile-centric applications in public and industrial sectors, extending beyond location-based services to mobility management, resource management, and user-centric applications. The arrival of Fifth Generation Mobile Networks (5G) technologies enables sub-meter positioning accuracy in outdoor scenarios, but a global and unified solution is still missing in Global Navigation Satellite System (GNSS)-restricted situations. Cloud, fog, or network-based localization methods proposed in recent studies, such as [1], require a continuous network connection, and therefore any connectivity loss results in simultaneous localization failure. End-user devices or User Equipment (UE)s, such as mobile phones, wearables, or Internet of Things (IoT) devices are often limited in their performance by battery limitations, network accessibility, or other computational constraints. Therefore, reducing the computational, storage, and network requirements is essential to run efficient positioning algorithms on such devices [2]. Whether applied in an industrial complex, hospital, entertainment center, or shopping mall, finding fast and lightweight techniques for reliable localization is essential for asset security, user safety, Quality of Experience (QoE) and Quality of Service (QoS).

Utilizing radio signal measurements for indoor localization is widely applied across technologies, including IEEE 802.11 Wireless LAN (Wi-Fi), Bluetooth Low Energy (BLE), Ultra Wide-Band (UWB) or cellular network signals, while utilizing various techniques, such as propagation-based models, fingerprinting, or dead-reckoning [3, 4]. The signals used for localization range across signal strength measurements (RSS, Reference Signal Received Power (RSRP)), directional measurements (Angle of Arrival (AoA)) and temporal information (Time Difference of Arrival (TDoA)).

In a typical indoor environment, such as a factory, office complex, or university, the signal propagation is characterized by sparse Line of Sight (LoS) and strong multipath propagation, making the model-based localization techniques unreliable, just like directional or temporal signal measurements, which is why RSS measurements and non-parametric methods are utilized. In the scope of this work, we focus on RSS-based indoor positioning called fingerprinting as one of the most relevant indoor positioning methods [5, 6], which typically utilizes a K-Nearest Neighbors (K-NN) [7] model to estimate the UE position by finding the closest samples from the labeled training database (radio map). The volume and quality of the radio map determine the achievable performance, but

- *L. Klus was with the Department of Electrical Engineering, Tampere University, 33720 Tampere, Finland, and Institute of New Imaging Technologies, Universitat Jaume I, 12071 Castellón de la Plana, Spain. E-mail: lucie.klus@tuni.fi*
- *R. Klus, E. S. Lohan and J. Nurmi were with the Department of Electrical Engineering, Tampere University, 33720 Tampere, Finland.*
- *J. Torres-Sospedra is with the ALGORITMI Research Center, Universidade do Minho, 4800-058 Guimarães, Portugal.*
- *C. Granell was with the Institute of New Imaging Technologies, Universitat Jaume I, 12071 Castellón de la Plana, Spain.*

This work was supported by the European Union's Horizon 2020 Research and Innovation programme under the Marie Sklodowska Curie grant agreements No. 813278 *(A-WEAR: A network for dynamic wearable applications with privacy constraints, http://www.a-wear.eu/) and No.* 101023072 *(ORIENTATE: Low-cost Reliable Indoor Positioning in Smart Factories, http://orientate.dsi.uminho.pt) and Academy of Finland (grants #319994, #323244).*
Manuscript submitted November 19, 2021.

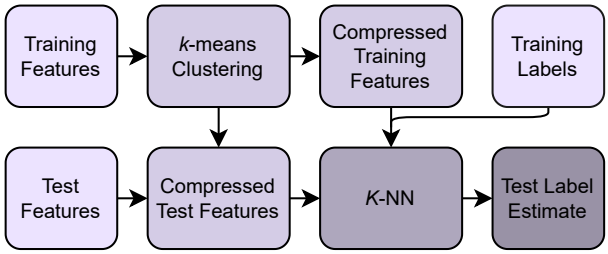

Fig. 1: Simplified block diagram of the proposed system, where EWOk is used to create the reduced radio map and K-NN estimates the user's position.

the more samples the radio map consists of, the slower and more computationally complex the positioning task is. Consequently, K-NN-based fingerprinting creates a trade-off between maximizing and minimizing the size of the radio map and its efficient utilization becomes challenging in large-scale deployments, as well as on performance-restricted devices [8].

To reduce the strain on the positioning device we propose and evaluate EWOk, an Element-Wise cOmpression using k-means, which reduces the size of the individual elements of the RSS radio map on the bit level while sustaining the database's positioning capabilities. The simplified implementation of EWOk to a K-NN-based positioning scheme is introduced in Fig. 1. EWOk, as a compression scheme, achieves a substantial reduction of the radio map data size, while leaving the number of samples and Access Point (AP)s in the dataset unchanged. Multi-dimensional compression can be achieved by combining EWOk with additional sample-wise compression schemes. EWOK performs initial compression and offline evaluation of the dataset positioning performance on the network side, while the end-user devices only perform localization on the reduced database in real-time. The proposed implementation of EWOk with K-NN positioning has the following advantages over the plain K-NN deployment:

- Due to the reduced radio map, the system effectively saves network data, as well as on-device storage.
- Adjustable trade-off between Compression Ratio (CR) and positioning error, which enables EWOk to adapt smoothly to deployment requirements.
- A faster operation of the fingerprinting models while using K-NN, especially on voluminous datasets.

The main contributions of this paper are as follows:

- We implement EWOk, an Element-Wise cOmpression using k-means, as an effective RSS radio map compression technique applicable on fingerprinting datasets. We show that all RSS data points can be stored using a 7-bit representation with negligible compression error, and that they can be further compressed into lower-bit representations using EWOk. Consequently, we propose an RSS-based Indoor Positioning System (IPS) with an offline training phase performed on the network, and prediction phase at the UE, while minimizing computational, memory, and data transfer loads.
- We propose 6 different initialization methods for k-means clustering based on the input data distribution, removing the effect of randomness from the resulting positioning performance. The initialization methods work on arbitrary data and are not limited to the proposed system.
- We apply and analyse EWOk with K-NN positioning on 25 different RSS positioning datasets (Wi-Fi, BLE, and simulated), and compare the localization performance before and after the compression, when utilizing both the simple configuration and the best-performing positioning algorithm settings of the K-NN, according to the known literature. We then further improve the compressed-datasets positioning performance by finding the optimum parameters for our implementation, resulting in improved positioning compared to the best-performing parameter results across all datasets.
- We demonstrate the multi-dimensional compression capabilities of EWOk by combining it with Principal Component Analysis (PCA) to achieve the combined compression of both feature-vector and individual data elements. We show, that the combined approach outperforms the standalone solution in terms of trade-off between the positioning accuracy and CR by a significant margin.

All the contributions mentioned above have a positive impact on extending the capabilities of the current on-device IPSs by enabling UEs to operate with larger, and therefore more robust and accurate positioning databases. The solution can be implemented across the spectrum of technologies and deployments, and ensures the efficient and uninterrupted localization regardless the connectivity status. Additionally, stand-alone EWOk might be applied on other (including non-positioning) kinds of data.

The rest of this paper is structured as follows: Section 2 presents the overview of the current State-of-the-Art and pinpoints the knowledge gaps, which are filled by this paper. Section 3 describes the methods and materials used in this paper. These are further utilized in Section 4, where the proposed method is explained in detail. Section 5 introduces the metrics used for the evaluation of the proposed system and presents the numerical results, followed by the Discussion subsection. Section 6 summarizes the main findings.

2 Related Literature

2.1 Fingerprinting-based Indoor Positioning

Fingerprinting localization is one of the most popular solutions of IPS, mostly as it does not require previous knowledge of the environment or location of APs on the side of the model. As a trade-off, its performance is strongly determined by the database of available fingerprints, namely the database's quality, granularity, and up-to-dateness. The State-of-the-art (SOTA) on indoor positioning is surveyed in [3, 4, 9, 10], discussing available solutions, algorithms, and technologies for IPS. These surveys list and evaluate localization techniques, used technologies, and/or applications of indoor localization in health, security, or tracking services.

The key challenge of fingerprinting is to ensure a stable environment with a high-quality radio map. The authors of [11] proposed a secondary BLE beacon deployment in areas poorly covered by Wi-Fi signal and proposed a hierarchical system to perform localization. The authors achieved good positioning accuracy, but the additionally needed infrastructure could prove unfeasible in certain cases (cost, restrictions, etc.). To cope with the rapid changes within the localization environment, such as AP movement or power adjustment, the authors of [1] proposed an automatic fingerprint update algorithm to filter out any outdated APs in the environment by using Gaussian process regression. The work proposed that the UE localization and database update are performed at the server side, which, on one hand, reduces the computational load for the UE, but, on the other hand, disables the localization if the connection link is lost. In contrast, our proposed scheme does not reduce the number of APs, nor the number of measurements. The authors of [12] built an IPS without performing a site survey from the Full Model (FM) signal distributions. They proposed a model based on public data about base stations' locations to obtain the radio map by using a path-loss model. The localization is then performed using K-NN method and path-matching with promising results. Another Wi-Fi localization system without the requirement of prior site survey was designed in [13]. Tilejunction model proposed in [14] utilizes a linear programming approach to mitigate the noise contained within Wi-Fi fingerprints for accurate localization by matching the results to created tiles, rather than the training fingerprints. The presented results showed an improved positioning performance over the benchmark methods, such as Kullback-Leibler divergence-based method or RADAR [15]. The low-overhead fingerprinting system proposed in [16] reduces the implementation overheads by region-partitioning the APs in the deployment. The evaluation performed with heterogeneous devices over a long period showed the method's robustness. Similar conclusions were found in [17], implementing a self-updating algorithm for RSS samples. The authors of [18] propose a novel matching algorithm for localization that considers spatial relations between the samples on top of their similarity in feature space.

2.2 Boosting the Performance of K-NN Fingerprinting

Much research has focused on improving the K-NN's performance by utilizing additional algorithms or physical quantities [5]. For instance, combining RSS, magnetic, and motion data can highly increase the quality of the crowdsourced fingerprinting database, as well as that of the prediction itself. The $UbiFin$ system proposed in [19] mitigates signal bias and path error while mapping both Radio Frequency (RF) and magnetic data into the training database. The presented results outperform the stand-alone RSS solution by a significant margin. When performing prior clustering, as proposed by [20], their algorithm is able to boost the fingerprinting prediction speed. Specifically, it narrows the K-NN search space to the fingerprints with the same strongest AP as a reference measurement. Consequently, the improvement in the prediction speed leads to a decrease in positioning accuracy.

Numerous works aim to improve the performance of K-NN by optimizing the algorithm itself by e.g. weighting samples or features. A two-fold, Weighted K-NN algorithm is proposed in [21], where in the first iteration the algorithm selects the closest cluster of fingerprints, and it finds in the second iteration the positioning estimates from searching in the selected cluster's samples. The method boosts prediction time at the cost of positioning accuracy.

In this work, we focus on improving the performance of K-NN by combining it with additional methods (clustering and PCA). As a side note, the utilized code performs matrix-based distance search and task parallelization, which boost the prediction speed compared to the plain algorithm but are not the main research objectives of this work.

2.3 RSS Radio Map Compression

The initial idea of utilizing k-means clustering as a compression method was previously presented by the authors in [6]. The work introduced an offline compression scheme with an online adaptive loop, which allows datasets to update over time and therefore is able to adjust to a slowly changing environment. Although the resulting adaptive algorithm does not decrease positioning performance, it assumes all online fingerprints as trustworthy and indirectly incorporates them into the training dataset. The work simplifies the setting of parameter k depending on the number of unique values in the training data only, leading to sub-optimal settings for certain datasets, resulting in higher errors at the same compression level. The uncertainty of the random initialization is not considered as well.

The authors of [22] combine the floor-wise k-means clustering with K-NN algorithm to significantly reduce the radio map and the floor prediction time, compared to the standard K-NN approach. The proposed model extracts several representative centroid heads per floor, which are later used to estimate the floor. The resulting floor hit rate is comparable with the benchmark method.

The topic of radio map compression while boosting the performance capabilities was also broadly covered by [23]. The fingerprinting dataset is transformed into a radio map image, which is compressed using Discrete Cosine Transform (DCT). This method allows significant size reduction while its positioning capabilities are comparable to the traditional fingerprinting approach. The disadvantage of utilizing DCT to compress the radio map is the necessity to perform inverse DCT to recover all fingerprints before utilizing the data further.

In contrast to the literature presented above, we propose a lightweight compression method that can be implemented into any existing mobile system that boosts the desired system's data storage and transfer capabilities. Additionally, we consider 25 different indoor positioning datasets previously used in the literature for evaluation, rather than considering only a single, convenient deployment, in order to show the wide and unrestricted applicability.

3 Materials and Methods

In this section, we introduce the algorithms, parameters and datasets that contribute to the proposed solution. The

symbols and notations used in the paper are summarized in Table 1. For the sake of clarity, we denote the number of clusters of k-means as k, while the number of considered neighbors of K-NN is represented by capital K.

TABLE 1: Symbols and notation used in this paper

AP	Number of APs [-]
CR	Compression ratio [-]
CR_{EWOk}	Compression ratio of EWOk [-]
CR_{pca}	Compression ratio of PCA [-]
CR_{tot}	Compression ratio of combined compression methods [-]
$\Delta_{\epsilon_{3D}}$ Δ_{ζ}	Dissimilarity parameter [-]
ϵ_{3D}	3D positioning error [m]
$\tilde{\epsilon}_{3D\alpha}$ $\tilde{\epsilon}_{3D\beta}$	Normalized 3D positioning error to baseline α, β resp. [-]
K	Number of neighbors in K-NN [-]
k	Number of clusters in k-means [-]
$\mathcal{O}$	Complexity [-]
S	Number of samples in the whole dataset [-]
S_{test}	Number of samples in the testing dataset [-]
S_{train}	Number of samples in the training dataset [-]
Thr	Threshold for total variance of PCA [%]
ζ	Floor hit rate [%]
$\tilde{\zeta}_{\alpha}$ $\tilde{\zeta}_{\beta}$	Normalized floor hit rate to baseline α, β resp. [-]

3.1 K-Nearest Neighbors algorithm

The K-NN algorithm is one of the most commonly used indoor positioning methods, especially in the context of fingerprinting approaches [3, 7, 12]. The algorithm requires the existing (training) database of fingerprints consisting of features (RSS measurement array) and the corresponding labels (positioning coordinates, building, and floor indexes). To estimate the labels of a new sample, it calculates its distance to each sample's features from the training database based on the specified distance metric. For K-NN algorithm, the training dataset is not used to train the specific weights or parameters of the model, as is the case with Neural Network (NN), Support Vector Machine (SVM) or other Machine Learning (ML) algorithms. Here, the training database serves directly as a source of samples that are used to predict the currently considered labels. As such, the plain version of K-NN does require no training, but as a trade-off, prediction is usually more resource-expensive than in the other methods, especially if the training dataset is voluminous. The lack of training phase for K-NN is often considered an advantage since there is no risk of poorly training the model, which can occur when using ML methods. Despite K-NN's drawbacks and limitations, it is still one of the most efficient, accurate and well-performing algorithms used for indoor positioning purposes [24, 25].

In terms of complexity, the training phase of K-NN is described as $\mathcal{O}(1)$ as no prior training is required, while the complexity of prediction is generally defined as

$$\mathcal{O}(S_{test} \cdot K \cdot AP) \tag{1}$$

depending on the size of the vocabulary (num. of training samples), the number of considered neighbors, and the dimensionality of the input. Moreover, the complexity of K-NN is dependent on the selected distance metric.

Much research in the related literature has resulted in numerous extensions and alterations of the K-NN. Weighted K-NN (WKNN) and its alternatives [26], authors in [27] additionally consider the importance of chosen nearest neighbors by the inverse of their distance, which in certain cases leads to improved performance. The optimization of K-NN's prediction time by applying clustering is widely described in Section 2. The authors of [28] propose the kTree method to choose the optimal number of neighbors K without the costly cross-validation. In this work, we utilize the plain version of K-NN.

3.2 k-means Algorithm

One of the fundamental building blocks of the proposed compression algorithm is the utilization of k-means clustering algorithm [7] to reduce the number of possible values in the RSS data. Consequently, the allowed values are based on the data distribution of the specific dataset, minimizing the resulting reconstruction error caused by the compression. As a result, we are able to represent each value from the whole RSS dataset using a smaller number of bits, as we described below.

Despite the k-means clustering algorithm being one of the basic clustering approaches, careful choice of its hyperparameters and behavior is crucial in order to maximize performance. The first and foremost parameter of the method is the selection of the number of clusters, denoted as k. In k-means, each cluster is specified solely by its centroid coordinates, and the final k is selected most commonly by parameter sweeping. The proposed k-means compression in EWOk is based on substituting the values of the RSS data in the dataset with the coordinate of their closest centroid (a single number).

The second parameter, the distance metric, defines how the similarity between each sample and the centroids is calculated. In addition, other parameters and configurable functions included in the algorithm are defining the iterative behavior, the means of centroid initialization before the first iteration, the action after finding the empty centroid, the maximum number of iterations, convergence definition, number of replicates, and more.

The k-means is initialized by selecting k initial clusters according to the pre-defined initialization method. Next, the algorithm repeats the following two steps until convergence. First, each input sample is assigned to its closest centroid based on the distance metric. Second, the centroid coordinate is adjusted to minimize the distance to all its assigned samples. The algorithm finishes after the centroid coordinates do not change between two iterations (convergence) or the maximum number of iterations is reached.

We utilize k-means, rather than other, more complex clustering algorithms since our goal is to define each cluster by a singular value in order to perform efficient compression. Compared to Gaussian mixture model clustering, which defines each cluster centroid by its center coordinate and its covariance, k-means is much faster to train since it does not have to fit the distributions in each iteration. One of k-means' advantages is its linear complexity of training defined as

$$\mathcal{O}(n \cdot k \cdot d \cdot i) \tag{2}$$

where n determines the number of d-dimensional samples, k represents the number of clusters and i the number of re-

quired iterations to converge [7]. For EWOk, the complexity is defined as

$$\mathcal{O}(S_{train} \cdot AP \cdot k \cdot i) \tag{3}$$

as the inputs are one-dimensional and the number of input samples equals $S_{train} \cdot AP$. The k-means' downside of being able to represent only symmetric shapes is diminished by the fact that we consider single numbers as inputs. Other methods, e.g. density-based clustering methods, are unsuitable for the task.

3.3 Data Representations and Distance Metrics

As described above, both k-means and K-NN algorithms measure the similarity between samples based on their calculated distance. As a result, two separate parameters are implemented and applied on the data within the system, namely data representation and distance metrics.

Signal strengths are traditionally measured in decibel-milliwatts (dBm), and the difference of 3 dBm means a double increase or decrease of the signal strength measured in Watts. This example clearly shows that the choice of the units in which we represent the data has a large impact on the resulting differences between two samples. The "units" in which we represent the data are specified by the data representation parameter. We consider 3 data representation options: positive, powed with β = e, and exponential with α = 24, as defined in [29]. Positive data representation is a linear transformation, which subtracts the minimum value from the database from all samples and represents the unmeasured APs by 0. Powed and exponential representations introduce non-linearity to the measurements, which improve later positioning performance of certain datasets [5].

After turning the data into the desired format by changing their data representation, it is necessary for both k-means and K-NN algorithms to calculate the distances between the samples. In this work, we utilize 9 different distance metrics for K-NN, namely Manhattan, Euclidean, Squared Euclidean, Hamming, Logarithmic Gaussian Distance (LGD), Neyman, Penalized Logarithmic Gaussian Distance with penalty 10 (PLGD10) and 40 (PLGD40), and Sørensen [29], [30],to optimize the performance of K-NN positioning. The selected distances were chosen from numerous alternatives based on their performance and applicability in the related literature.

For k-means clustering, only Manhattan and Squared Euclidean distance metrics were considered. When utilizing Manhattan distance, compared to the Squared Euclidean, the samples further from the centroid have lesser impact on the result, while samples closer to it have a stronger impact on the coordinates of the centroid. Consequently, the centroid selection is more affected by the "close samples' majority vote". The remaining distances are found unsuitable for the task due to e.g. their regularization parameters.

3.4 PCA

PCA is an algorithm, which extracts the principal eigenvectors from the multi-dimensional data and uses them to transfer the data into their orthogonal basis [31]. It is usually calculated using Singular Value Decomposition (SVD) algorithm and is often utilized for data compression, dimensionality reduction [32], or feature extraction [33] as stand-alone solution or combined with other methods.

We utilize PCA as the compression scheme with the adjustable CR mechanism based on the desired total variance (denoted as threshold or Thr) that is meant to be preserved within the data. After calculating all principal component coefficients, we only choose the N strongest eigenvectors, within which the desired total variance is included. The principle of PCA is widely covered in the referenced literature, therefore we omit the detailed mathematical description.

3.5 Available Datasets

In this work, we utilize 25 fingerprinting datasets in order to evaluate our proposed methods, and compare them to other previously published works. These datasets were created by University of Minho, Portugal (DSI 1&2 [34], MINT 1 [35]), Universitat Jaume I, Spain (SIM 1 [5], UJI 1&2 [36], UJIB 1&2 [37], and LIB 1&2 [38]), University of Extremadura, Spain (UEXB 1&2&3 [39]), University of Mannheim, Germany (MAN 1&2 [40], [41]), University of Sydney, Australia (UTS 1 [42]) and Tampere University, Finland (TUT 1&2 [22, 30], TUT 3&4 [43], TUT 5 [44], TUT 6&7 [45], SAH 1 and TIE 1 [46]). Additional and detailed information about the majority of the datasets may be found in [5], including the SIM 1 dataset. Moreover, we choose the fingerprinting datasets gathered using multiple technologies, namely Wi-Fi (DSI 1&2, LIB 1&2, MAN 1&2, MINT 1, TUT 1-7, UJI 1&2, UTS 1), BLE (UJIB 1&2, UEXB 1&2&3) and simulated environment (SIM 1), to demonstrate the universal applicability of the proposed solution. Some or all of these datasets were previously used in many other publications including (but not limited to) [26, 47].

4 Proposed System Model

4.1 General System Model

Below we specify the individual components of the considered indoor positioning scheme, visualized in Fig. 2. The proposed Element-Wise cOmpression using k-means, or EWOk, includes the k-means clustering of the training features, the creation of the reduced training database, and the compression of new samples. The considered positioning prediction (with the K-NN algorithm) is performed after EWOk. We denote, that K-NN can be interchanged for an arbitrary positioning algorithm as it works independently with the EWOk scheme.

In order to off-load the majority of the computational load to train and evaluate the model from the UE to the network side, the proposed system model is divided into offline and online stages. The offline training is realized on the network side, and its main objectives are to find the representative centroid coordinates from the training data, compress the original radio map and evaluate the performance of the system while tuning the system parameters such as k, initialization method, K in K-NN or the distance metrics. Online prediction is realized on the UE's side and its only objective is to accurately estimate the device's location. In practice, to enable the online prediction, the UE requires the reduced radio map and the centroids.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2023.3277333

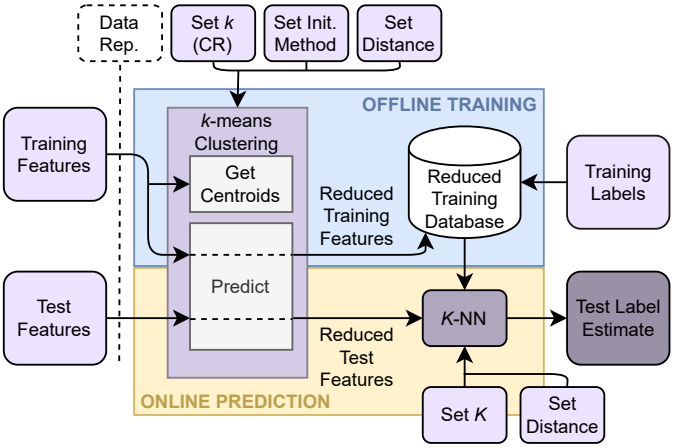

Fig. 2: General System Model, including Offline Training and Online Prediction phases.

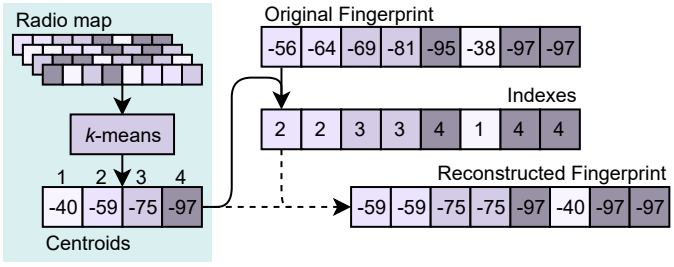

Fig. 3: Simplified EWOk compression flow with an example of a single fingerprint compression.

Fig. 2 depicts the overall system model along with the most impactful parameters of each building block. The **offline training** is initiated by applying the chosen data representation on the training features (of the training dataset). Then, the EWOk algorithm is initiated. As shown in Fig. 2, the main parameters defining the EWOk's behavior are the chosen number of clusters k, which directly sets the achieved CR, initialization method, which is further discussed below, and the distance metric. The data fed to the k-means is the matrix of all training features, reshaped into a single, 1-dimensional vector. The algorithm returns the coordinates of the centroids and the clustered feature vector. The matrix of reduced training features is then created by reshaping the clustered vector to the original matrix shape. In order to create the reconstructed radio map, the centroid indexes are substituted with the corresponding centroid coordinates. The reduced training database is created by pairing the reduced training features with the corresponding labels (positioning coordinates). The simplified example of k-means training and later compression of a single fingerprint (AP=8) is depicted in Fig. 3 with $k = 4$ clusters and without applying data representation on RSS data for better visualization.

The **online prediction** is performed sample-wise on the side of the UE. First, the data representation is applied onto the sample, after which EWOk algorithm substitutes all values in the measurement array with the closest centroid coordinate. Afterward, the K-NN algorithm estimates the corresponding location by matching the reduced measurement array with the reduced training database. The behavior of K-NN regressor is defined by the chosen number of considered neighbors K and the selected distance metric [5]. Apart from that, the algorithm averages the neighbors' labels in case of equal distance from the sample when exceeding the chosen K, along with additional supporting functions ensuring the seamless flow of data.

4.2 Compression Efficiency of RSS Data

In this work, we consider the CR metric as the ratio between the original and compressed size of the radio map as:

$$CR = \frac{size(original\ radio\ map)}{size(compressed\ radio\ map)} \tag{4}$$

where $size()$ denotes the size used to represent the considered radio map. Therefore, CR = 3 denotes the three-fold decrease of the radio map size. Since the number of samples is unchanged throughout the compression process, the interpretation can be simplified to the ratio of sizes of a single measured RSS sample before and after its compression.

In order to objectively evaluate the compression capabilities of the algorithm, which compresses every individual RSS value, we first define the appropriate benchmark for the CR metric. According to the Institute of Electrical and Electronics Engineers (IEEE) 802.11 wireless Local Area Network (LAN) standard on radio resource measurements [48], ETSI EN 300 328 [49] and ETSI EN 302 502 [50] specifications, the maximum Wi-Fi antenna transmit power is 20 dBm for 2.4 GHz bands and up to 30 dBm in 5 GHz bands. The highest possible detectable Wi-Fi RSS values are approx. 10 dBm. Furthermore, the noise floor of the Wi-Fi signal is approx. -100 dBm, depending on the device, therefore lower RSS does not have to be considered. Moreover, the network reports of RSRP within Long Term Evolution (LTE) system map the measured signal strength into 113 integer values, with the reporting range from -156 dBm to -44 dBm with 1 dB resolution, as defined in 3rd Generation Partnership Project (3GPP) standards [51], while the New Radio (NR) standards consider 128 values [52] instead. According to the current standards, the RSS values are reported as the whole numbers $\mathbb{Z}$, limiting their resolution. Consequently, the whole range of possible RSS values may be represented using 7 bits data format, since 7 bits are able to represent up to 128 different values. As the result, the benchmark and the CR value of 1 (no compression) refers to the raw RSS values with 7-bit representation.

Nevertheless, the measured RSS values in datasets MAN 2, TUT 1, TUT 2, TUT 5, MINT 1, UEX 1, UEX 2, UEX 3, UJI B1, and UJI B2 were post-processed by means such as averaging or interpolating the measurements over a predefined area. As an outcome, the RSS values in these datasets are stored in 64 bit (double) format (values belong to a subset of real numbers $\mathbb{R}$). In our previous work [6], we considered 64 bit representation as the benchmark for such data. We prove that the highly accurate data format of the RSS values is redundant in Section 5 and that such data can be equivalently represented using 7 bits only. As such, all RSS values in all datasets are transformed into the 7-bit representation and thus we are able to define the common baseline for the CR. Nevertheless, the true CR of real-valued

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2023.3277333

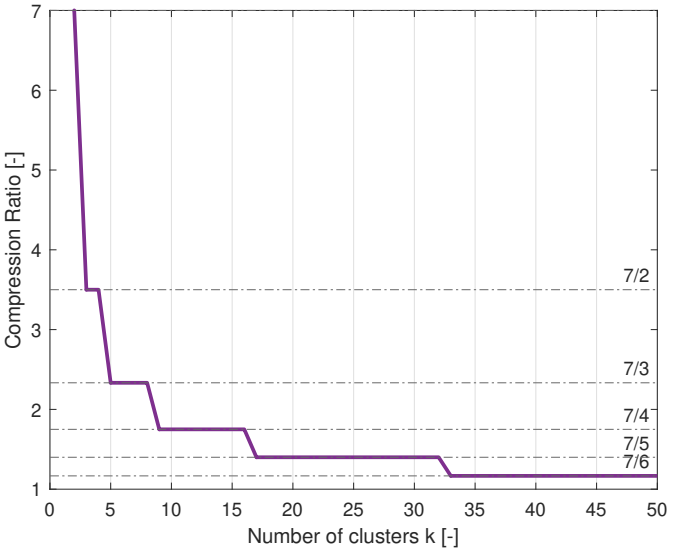

Fig. 4: Achievable CR using EWOk based on the number of clusters (possible values) in the data.

datasets is considerably higher, as we reduce their bit-wise representation from 64 bits, instead of just from 7.

As described above, EWOk performs the compression of each element in the radio map. Consequently, the obtained CR of EWOk towards the 7-bit baseline is calculated as:

$$CR_{EWOk} = \frac{7}{ceil(log_2(k))} \tag{5}$$

where $ceil()$ denotes a function rounding up to the nearest integer, and k is the number of clusters in k-means. We generalize the EWOk CR to the full radio map compression since the total number of elements during EWOk (number of APs and measurements) compression remains unchanged.

It is also possible to show the dependency of the CR_{EWOk} on the number of clusters k in the proposed method, as the number of clusters directly states the amount of possible RSS values across the whole compressed dataset. Fig. 4 visualizes such dependency and shows that the higher the compression ratio, the lower number of clusters, and therefore fewer bits are required to distinguish different RSS values.

Fig. 4 also shows that for the maximum CR and the highest possible number of clusters, it is necessary to choose the number of clusters k equal to the powers of 2, e.g. 2, 4, 8, or 16, since those refer to the maximum number of values stored using 1, 2, 3, or 4 bits, respectively. The CR is then calculated as stated above.

When the multidimensional compression involving PCA is considered, the CR calculation has to be adjusted accordingly. PCA reduces the number of APs in the dataset, this is why the resulting CR (CR_{pca}) is obtained as a ratio of the number of APs in the original dataset to the number of APs after the PCA compression. We then combine the two compression schemes, as described later. The resulting CR of the combined methods (CR_{tot}) is calculated as:

$$CR_{tot} = CR_{pca} \cdot CR_{EWOk} \tag{6}$$

In addition, when calculating the CR we consider only the ratio of the training and test feature sizes before and after compression. The size of both training and test labels is omitted in the calculation, as the compression is not applied there and its impact on the total size is different for each dataset. Additionally, we omit the additional overhead necessary to perform the positioning, namely the array of cluster centroids and, in the case of utilizing PCA, the coefficient matrix. Nevertheless, their size is insignificant compared to the size of each dataset.

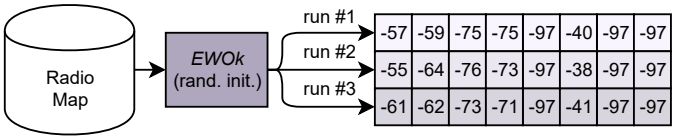

Fig. 5: Simplified random initialization example

4.3 Random initialization effect of k-means algorithm

The main aspect affecting the performance of the K-NN algorithm is the actual input data (given the same parameters), resulting in an identical outcome each time the algorithm is repeated. In contrast, the k-means algorithm in its default version randomly initiates the initial centroid coordinates and consequently converges to different final constellations. Fig. 5 demonstrates such behavior, depicting three different runs of EWOk with the same settings and training data while resulting in different centroid coordinates. As a result, the reconstruction error of the RSS data, as well as a resulting positioning performance while utilizing the reduced dataset may vary after each run. The issue of k-means random initialization effect was extensively studied in [53], where the authors highlight the importance of proper initialization algorithm. Moreover, the survey states that the most reliable way to find the true cluster centroids is by repeating the algorithm, which creates additional training overhead.

The initialization of the algorithm also determines the number of iterations that the algorithm needs to perform before convergence, as introduced in Eq. 2, effectively determining the algorithm's complexity. In this work, we evaluate two distinct random initialization algorithms combined with EWOk, namely random sample initialization and *k*++ initialization. Random sample initialization, further denoted as "random", initiates the centroid locations by drawing k different samples at random from the input data. The *k*++ initialization, proposed by [54], is a randomized version of "Furthest point heuristic" algorithm [55]. The *k*++ selects the first centroid at random from the training samples' population, and each subsequent centroid is chosen as a random training sample with the probability proportional to the sample's distance from the currently chosen centroids. The method increases both convergence speed and accuracy of k-means. Nevertheless, due to the randomness of the initialization method, the final result after each run may significantly vary and numerous repetitions of the algorithm have to be performed in order to find the desired solution.

In case the evaluation metric is the error between the original feature vector (all training samples' features reshaped into a single vector) and its reconstruction after EWOk, the resulting performance after each iteration can

be easily calculated, and the best-performing centroid selection can be selected directly. When applying the clustered dataset's 3-Dimensional (3D) positioning accuracy as the primary evaluation metric, as is the case in this work, the performance evaluation requires extra steps and effort. According to our experiments, lower difference between the original and reconstructed samples does not necessarily mean better positioning performance. Consequently, to evaluate the centroid selection after each iteration, the system has to perform K-NN positioning using the compressed dataset to obtain the value of the 3D positioning error. As such, the cost of evaluating and optimizing the solution substantially increases.

In this work, we propose several approaches to select the initial cluster coordinates for k-means algorithm in order to completely remove the "different run, different result" methodology from the initialization. Those approaches are derived from the training samples' distribution. The proposed initialization settings are based on the Empirical Cumulative Distribution Function (ECDF) of the input data and their goal is to always set a reasonable starting point for the clustering algorithm. The general idea behind all proposed settings is to divide the ECDF of the vector of training features into segments, whose borders are selected as the initial centroid coordinates. All initialization settings disregard the unmeasured values from the input vector since the majority of samples across all databases include more than half of their measurements as unmeasured values, which would consequently skew the ECDF. We propose "max", "min", "xtr", "imax", "imin", and "ixtr" initialization settings based solely on the input's distribution, from which the best performing one can be obtained while evaluating the training database.

The "max" and "min" initializations equidistantly divide the cumulative distribution function into N segments using $N-1$ horizontal lines, where N equals the number of clusters k. Thus, each segment contains approximately the same number of (measured) samples. The $N-1$ values at which the horizontal lines intersect the distribution are selected as the initial centroid coordinates. Additionally, "max" initialization sets the maximum measured value to the N^{th} cluster, whereas the "min" method assigns the minimum measured value - 1 to the N^{th} cluster (the value considered as the unmeasured in the dataset). The "xtr" setting equidistantly divides the distribution using $N-2$ lines, and the two remaining clusters are assigned to the minimum and the maximum, respectively.

The "imax" and "imin" settings (incremental max and min) divide the ECDF similarly, only the distances between the horizontal lines are linearly increasing. The ECDF is first divided into $\sum_{i=o}^{N-1}(N-i)$ segments, and starting from the top, the 1^{st} horizontal line spans 1 segment, the 2^{nd} line 2 segments, etc. The intersections of lines and the distribution curve are then chosen as the first $N-1$ centroid coordinates, and the last centroid is assigned to the maximum and minimum, respectively. The "ixtr" setting performs the division similarly into $\sum_{i=o}^{N-2}(N-i)$ segments and the first $N-2$ centroids are chosen accordingly. The two remaining centroids are assigned to the maximum and minimum value from the input vector. The individual initialization methods are depicted in Section 5.

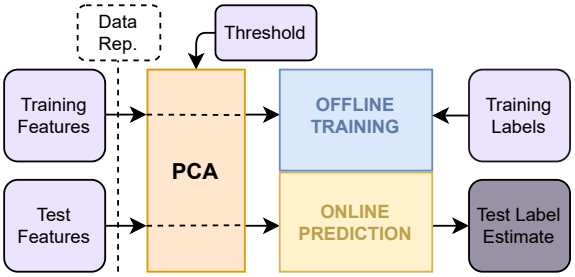

Fig. 6: Simplified system model with implemented PCA compression scheme applied after data representation. The threshold Thr defines the variance kept after compression.

4.4 Multidimensional Compression

In order to boost the compression capabilities of the presented system, and to achieve a true multidimensional compression, we additionally implement the PCA-based compression into the scheme in order to reduce the number of APs while minimizing the loss of information included within the data. Applying PCA results in deeper compression, improved prediction times, and in certain cases, improved positioning performance, as we will show in the following Section 5. The PCA compression is applied after applying the data representation onto the data in the scheme, as shown in Fig. 6 and before applying EWOk. The coefficients and the eigenvectors are obtained by performing the analysis on the training features only, and then they are applied to the test features. The rest of the compression scheme is left unchanged, and therefore EWOk is now applied to the resulting principal components' elements.

The only parameter we consider for PCA is the percentage of total variance left within the data, defined by the threshold (Thr). The same selection of Thr results in a varying number of principal components left per each dataset, and therefore the CR is different per dataset as well.

Additionally, as PCA reduces the number of elements in each feature vector, it reduces the complexity of the K-NN algorithm at the same time to

$$\mathcal{O}(S_{test} \cdot K \cdot \frac{AP}{CR_{pca}}) \tag{7}$$

as the number of APs is effectively reduced.

We implement the additional dimensionality reduction scheme to demonstrate EWOk's compatibility with other methods and the PCA applied prior to the EWOk can be freely changed to any other dimensionality reduction method, such as autoencoder, spectral embedding [56], or isomap embedding [57].

5 Evaluation and Numerical Results

In this section, we introduce the means of evaluation of the proposed model, including the evaluation metrics and used benchmarks. Further, we present the numerical results.

In order to ensure the repeatability, replicability, and reproducibility of our work, we provide all information

required to reproduce the experiment. We also provide the source code, which is available online on Zenodo [1].

5.1 Evaluation Metrics and Baselines

5.1.1 Evaluation Metrics

In order to objectively evaluate the proposed method along with all utilized algorithms, we implemented the following metrics.

Floor-hit, further denoted as ζ, evaluates the ability of the positioning algorithm, such as K-NN, to correctly establish the correct building and floor number for the given dataset. Floor hit is calculated as the percentage of correctly estimated samples (for both building and floor label) as:

$$\zeta = (\frac{1}{n}) \sum_{i=1}^{n} (bld_i == \overline{bld}_i \& flr_i == \overline{flr}_i) \cdot 100\% \quad (8)$$

where n is the number of samples, bld_i denotes the building index of the i^{th} sample, $\overline{bld}_i$ denotes the estimated building index of the i^{th} sample, flr_i denotes the floor index of the i^{th} sample and finally $\overline{flr}_i$ denotes the estimated floor index of the i^{th} sample.

We evaluate the positioning accuracy of the positioning algorithm using mean 3D positioning error ϵ_{3D}. 3D positioning error is calculated as the Euclidean distance between the coordinates of the original sample and the estimated sample. ϵ_{3D} is then the average error across all samples from the test dataset, as:

$$\epsilon_{3D} = \frac{1}{n} \sum_{i=1}^{n} \sqrt{\sum_{j=1}^{3} \left(y_{j,i} - \overline{y}_{j,i}\right)^2} \quad (9)$$

where j denotes the coordinate index, $y_{j,i}$ is the j^{th} coordinate of the i^{th} original sample and $\overline{y}_{j,i}$ is the j^{th} coordinate of the i^{th} sample's prediction.

Finally, we consider normalized values for all considered positioning metrics to better reflect the difference in performance between the baseline model and the proposed solution [58]. The considered metrics are normalized floor-hit $\tilde{\zeta}$, and normalized 3D positioning error $\tilde{\epsilon}_{3D}$. The normalized metrics are obtained as $\tilde{A}$ in:

$$\tilde{A} = \frac{A_{test}}{A_{baseline}} \quad (10)$$

where A_{test} stands for any of the evaluated results, namely ζ_{test} or $\epsilon_{3D,test}$, and $A_{baseline}$ stands for $\zeta_{baseline}$, or $\epsilon_{3D,baseline}$, respectively, and refers to the benchmark results obtained using the corresponding positioning baseline method α or β. Consequently, the 3D positioning error normalized to the α benchmark is denoted as $\epsilon_{3D,\alpha}$.

Normalized metrics directly compare the tested method's performance to the baseline. In case the resulting $\tilde{\epsilon}_{3D}$ is smaller than 1, the resulting positioning error is smaller than that of the baseline, e.g. $\tilde{\epsilon}_{3D}$ equal to 0.9 means that the method's 3D positioning error was decreased by 10%. As such, we aim to achieve $\tilde{\epsilon}_{3D}$ lower than 1. On contrary, we aim for $\tilde{\zeta}$ larger than 1 (as we aim to decrease the positioning errors and increase the floor hit).

1. The source code will be made publicly available on Zenodo after the paper is accepted.

TABLE 2: 64-bit vs. 7-bit dataset representation comparison

	64-bit representation		7-bit representation		64-bit vs 7-bit	
Dataset	ϵ_{3D}	ζ	ϵ_{3D}	ζ	$\tilde{\epsilon}_{3D}$	$\tilde{\zeta}$
MAN 2	2.47	100	2.40	100	0.97	1
TUT 1	9.59	90.00	9.59	90.00	1	1
TUT 2	14.37	72.73	14.37	72.73	1	1
TUT 5	6.92	88.39	6.96	88.29	1.01	1
MINT 1	2.67	100	2.70	100	1.01	1
UEX B1	3.71	90.20	3.66	90.20	0.99	1
UEX B2	4.65	94.20	4.65	94.20	1	1
UEX B3	7.14	76.67	7.30	78.33	1.02	1.02
UJI B1	3.05	100	3.03	100	0.99	1
UJI B2	4.33	100	4.28	100	0.99	1
Average					1	1

Additionally, we introduce a parameter Δ when evaluating the dissimilarity of two methods' normalized metrics, namely the dissimilarity of the normalized 3D positioning error as $\Delta_{\epsilon_{3D}}$ or normalized floor-hit as Δ_{ζ}. Given the normalized 3D positioning error of method A as $\tilde{\epsilon}_{3D}(A)$ and the normalized 3D positioning error of method B as $\tilde{\epsilon}_{3D}(B)$, their $\Delta_{\epsilon_{3D}}$ parameter is calculated as:

$$\Delta_{\epsilon_{3D}} = 1 - \frac{\tilde{\epsilon}_{3D}(A)}{\tilde{\epsilon}_{3D}(B)} \quad (11)$$

Consequently, $\Delta_{\epsilon_{3D}} > 0$ denotes the decrease of the normalized 3D positioning error of the method A, compared to method B by $\Delta_{\epsilon_{3D}} \cdot 100\%$. The Δ_{ζ} evaluating the normalized floor-hits is calculated similarly, and $\Delta_{\zeta} > 0$ denotes a lower floor-hit of the method A than that of the method B.

5.1.2 7-bit Benchmark

In this paper, we consider 7-bit representation as a benchmark for compression as described in Section 4. The stated CRs are calculated as if all datasets were represented by 7-bit formats, although some were originally represented by higher-bit representations (up to 64-bit), and therefore their actual CRs are up to $64/7$ times higher. These include datasets MAN 2, TUT 1, TUT 2, TUT 5, MINT 1, UEX 1, UEX 2, UEX 3, UJI B1 and UJI B2. The rest of the datasets are originally in integer format which can be transformed to 7-bit without the loss of data resolution.

To demonstrate the RSS dataset's positioning capabilities are not degraded by transforming the data from 64-bit to 7-bit representation, we first evaluate the positioning performance of the 64-bit datasets in their original data format. Next, we transform the RSS values in the above datasets into the 7-bit data format (represented by integer values obtained from rounding the original data) and evaluate the positioning accuracy of the transformed dataset. For both cases, we utilize a plain K-NN algorithm with K equal to 1, Manhattan distance metric and positive data representation. The precise results of the evaluation show close-to-equal positioning performance in both cases (Table 2).

Table 2 also proves that the performance of the 64-bit dataset is almost identical to that of the 7-bit dataset in terms of both the 3D positioning error and the floor-hit ratio. As such, we concluded that all RSS values in the datasets using 64-bit format can be reduced to 7-bits without any loss in

positioning accuracy. Moreover, the smaller data size allows for faster data processing and more efficient storage.

5.1.3 Benchmark results and database parameters

To fairly and unambiguously evaluate the impact of EWOk on the positioning performance, we utilize two positioning benchmarks for the evaluation. The first baseline, "Simple Configuration" or α, refers to results obtained while evaluating each dataset with the K-NN set to $K = 1$, Manhattan distance metric and positive data representation. The second baseline, "Best Coefficient" or β, follows the best parameter settings for each dataset from [5], using which the plain K-NN achieved the lowest positioning error. As 9 of the considered datasets were not included in [5], their "Best Coefficient" benchmark performance was obtained by performing the full parameter sweep, as described in the aforementioned work.

Table 3 includes the overview and the performance of all 25 considered databases, and lists the total number of samples in each database S, the number of training samples S_{train}, the number of test samples S_{test}, the number of APs and the type of wireless technology on which the databases were measured. The table then lists the 3D positioning error ϵ_{3D}, and floor-hit ζ of each dataset when evaluating the positioning performance using both baseline configurations (α and β). We selected a wide range of indoor positioning datasets, using different base technologies, different granularity of measurements and different density of APs in order to perform the analysis in different deployments.

5.2 Random vs. Non-Random Initialization

We evaluate the impact of random initialization (as explained in Section 4) on the resulting positioning accuracy across datasets and compare its performance across multiple repetitions with the proposed initialization methods, which require only a single run of EWOk.

Fig. 7 depicts the comparison of the positioning accuracy results between two random initialization methods, namely random sample initialization and *k*++ [54], along with the result of max initialization as the example non-random initialization defined later in the text. The figure presents the results for the number of clusters k from 2 to 25 and shows, that from the two random initializations, *k*++ is able to achieve better positioning accuracy despite the higher variance of the result. Fig. 7 additionally shows, that the variance of the results strongly differs across the individual runs of the algorithm and that in order to obtain the favorable result it is necessary to repeat the algorithm multiple times. On top of that, we show that the non-random initialization is able to achieve comparable results to the expected result of *k*++ without introducing uncertainty, and outperforms random sample initialization across the sweep with only a single repetition of the algorithm.

The results in Fig. 7 were obtained by running the proposed system with Manhattan distance for k-means, and K-NN with $K = 1$ and Manhattan distance metric. Each algorithm setting was repeated 100 times for each dataset, and the resulting positioning accuracy was normalized with the corresponding Simple Configuration (α) baseline. Each box in the resulting boxplot shows the median, 50% and 95% confidence interval of the sorted positioning results averaged across all databases.

Additionally, we show in Table 4 the mean number of iterations of the k-means algorithm performed before convergence for randomly initialzied algorithm, *k*++ initialization and the proposed max initialization. The results are aggregated across all 25 datasets and show that the proposed initialization method requires a significantly lower number of iterations than the randomly initialized algorithm, effectively reducing the complexity of k-means by minimizing the number of required iterations i, as introduced in Eq. 2.

As described in Section 4, we propose 6 non-random initialization methods that offer reasonable starting points for k-means. Fig. 8 depicts the initial centroid settings, as well as the centroid coordinates after clustering for the proposed initialization schemes on the dataset DSI 1 with $k = 4$. The lines mark the distribution points according to the initialization setting, and the selected centroid values are the RSS values at the intersections of lines with the dataset's ECDF. The figure shows that all 6 different initializations result in 6 different, although similar final centroid settings.

In Fig. 9 we present the performance of the individual proposed initialization schemes. The figure visualizes the normalized 3D positioning error $\tilde{\epsilon}_{3D_\alpha}$ of the compression towards the α benchmark (both with the same K-NN parameters) with the most compression-efficient number of clusters ($k = 8, 16, 32$). The results on the horizontal axis present the performance of each initialization and k setting per dataset, and clearly show that the compression scheme improves the positioning performance of certain datasets (UJI 1, TUT 7), and worsens the performance of others (MAN 1, UE B3). In some cases, the initialization setting defines, whether $\tilde{\epsilon}_{3D_\alpha}$ is improved or not (SAH 1, UTS 1). The last row of results presents the aggregated $\tilde{\epsilon}_{3D_\alpha}$ across all datasets as the representative metric, proposed in [58].

Fig. 9 shows that there is no single, best-performing initialization method. As a result, we propose to repeat the algorithm once with each setting during offline training and choose the best-performing one as a part of the system validation. Despite the proposed approach forcing the algorithm to repeat up to 6 times, it still drastically decreases the number of required repetitions and the variance of $\tilde{\epsilon}_{3D_\alpha}$ compared to the random initialization approach.

The rest of this work presents the results obtained using the proposed, non-random initialization schemes while considering either all of them or only max initialization as the representative method in cases where the evaluation does not consider parameter sweeping.

5.3 Numerical results of EWOk

In this section, we evaluate the performance of the proposed method with the best-performing models on each of the considered datasets. As the baseline for the comparison, we consider the β (Best Coefficient) K-NN setting, as found in [5], which obtained the best positioning performance across the performed in-depth parameter sweep. In order to impartially evaluate the impact of the k-means compression on the resulting positioning performance, we first performed a single repetition of the clustering with Manhattan distance metric and max initialization, while applying the K-NN with β parameters for positioning.

TABLE 3: Dataset Information and Baselines

	Dataset Information					α - Simple Config.			β - Best Coef. K-NN [5]					
Dataset	S	S_{train}	S_{test}	APs	technology	ϵ_{2D}	ϵ_{3D}	ζ	data rep.	distance	K	ϵ_{2D}	ϵ_{3D}	ζ
DSI 1	1717	1369	348	157	Wi-Fi	4.95	4.95	100	pow	Sørensen	11	3.79	3.79	100
DSI 2	924	576	348	157	Wi-Fi	4.95	4.95	100	pos	PLGD10	9	3.80	3.80	100
LIB 1	3696	576	3120	174	Wi-Fi	3.01	3.02	99.84	pos	Euclidean^2	11	2.46	2.48	99.94
LIB 2	3696	576	3120	197	Wi-Fi	4.02	4.19	97.72	pos	PLGD10	9	2.27	2.27	99.97
MAN 1	14760	14300	460	28	Wi-Fi	2.82	2.82	100	exp	Manhattan	11	2.06	2.06	100
MAN 2	1760	1300	460	28	Wi-Fi	2.40	2.40	100	exp	Neyman	11	1.86	1.86	100
SIM 1	11710	10710	1000	8	simulated	3.16	3.16	100	exp	Euclidean^2	11	2.41	2.41	100
TUT 1	1966	1476	490	309	Wi-Fi	8.61	9.59	90.00	pos	PLGD40	3	4.23	4.45	95.51
TUT 2	760	584	176	354	Wi-Fi	12.66	14.37	72.73	pow	Sørensen	1	7.80	8.10	92.05
TUT 3	4648	697	3951	992	Wi-Fi	8.92	9.59	91.60	pos	Sørensen	3	8.17	8.55	91.42
TUT 4	4648	3951	697	992	Wi-Fi	6.11	6.36	95.27	pos	PLGD10	3	5.07	5.40	95.98
TUT 5	1428	446	982	489	Wi-Fi	6.41	6.96	88.29	pos	PLGD40	3	5.25	5.26	99.59
TUT 6	10385	3116	7269	652	Wi-Fi	1.94	1.94	99.99	pos	Sørensen	1	1.90	1.91	99.99
TUT 7	9291	2787	6504	801	Wi-Fi	2.13	2.69	99.02	pos	Sørensen	1	2.06	2.24	99.31
UJI 1	20972	19861	1111	520	Wi-Fi	7.70	10.81	87.67	pow	Sørensen	11	6.17	6.56	95.23
UJI 2	26151	20972	5179	520	Wi-Fi	7.73	8.05	85.35	exp	Neyman	11	5.60	6.09	91.37
MINT 1	5783	4973	810	11	Wi-Fi	2.70	2.70	100	pow	PLGD10	11	2.16	2.16	100
SAH 1	9447	9291	156	775	Wi-Fi	8.16	9.07	46.80	exp	Neyman	11	6.03	7.20	44.23
TIE 1	10683	10633	50	613	Wi-Fi	4.25	7.16	60.00	pos	PLGD40	11	2.22	4.95	90.00
UEX B1	519	417	102	30	BLE	3.46	3.66	90.20	exp	Neyman	3	2.97	3.09	93.14
UEX B2	690	552	138	30	BLE	4.40	4.65	94.20	pos	Euclidean^2	3	4.19	4.31	97.10
UEX B3	300	240	60	30	BLE	6.59	7.30	78.33	pos	Euclidean^2	3	6.67	6.73	65.00
UJI B1	1632	732	900	24	BLE	3.03	3.03	100	exp	Neyman	11	1.64	1.64	100
UJI B2	816	576	240	22	BLE	4.28	4.28	100	pos	LGD	11	2.53	2.53	100
UTS 1	9496	9108	388	589	Wi-Fi	7.75	8.74	92.78	exp	Neyman	11	6.48	7.01	91.24

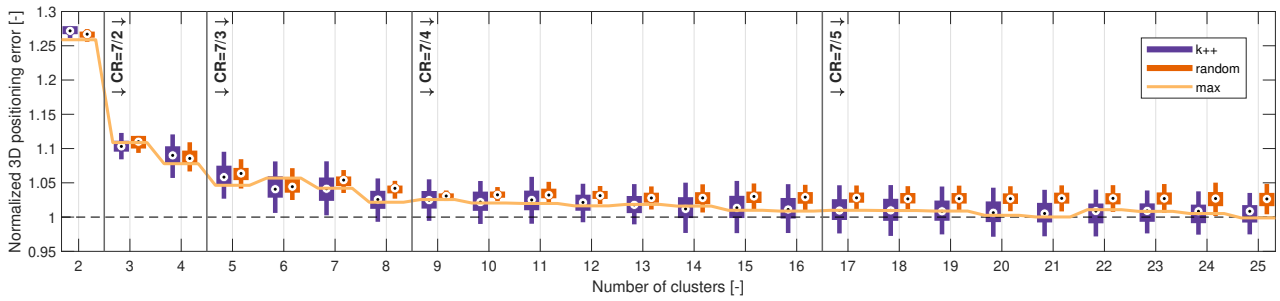

Fig. 7: Visualization of resulting 3D positioning error uncertainty after EWOk with *k*++ and random sample initializations, along with the non-random initialization max. The corresponding CRs are indicated by the vertical lines.

TABLE 4: The mean number of iterations of k-means before convergence for different initializations

Initialization	random	$k++$	max
Mean num. of iterations	27.0	7.5	5.1

Table 5 presents the normalized 3D positioning error $\tilde{\epsilon}_{3D_\beta}$, and the normalized floor-hit $\tilde{\zeta}_\beta$. Table 5 displays the results for the number of clusters k equal to 8, 16, and 32, all maximizing the number of clusters at their corresponding CR. Similarly to Fig. 9, the positioning performance of certain datasets is improved ($\tilde{\epsilon}_{3D}$ smaller than 1 and floor-hit $\tilde{\zeta}$ larger than 1; dataset TIE 1), or degraded (dataset TUT 2). Table 5 additionally shows the aggregated $\tilde{\epsilon}_{3D}$ and $\tilde{\zeta}$ over all datasets. On average, the 8-means setting increases the $\tilde{\epsilon}_{3D}$ by 5% while reducing the size of the radio map by 57.1%, 16-means by 2% with 42.9% reduction, and 32-means by only 1% with 29.6% radio map reduction, compared to the benchmark. The results show a negligible increase in positioning error and a relevant decrease in requirements for storage and energy savings.

Next, we performed a full parameter sweep across all 25 datasets, k-means distances and initializations, and K-NN parameters in order to find the best-performing settings for the compression scheme. The sweep was realized over 3 data representations, 6 k-means initialization methods, 2 k-means distance metrics, 1 to 35 Ks for K-NN, and 9 K-NN distance metrics (for details see Sections 3), resulting in 11 340 repetitions per dataset while considering only the single number of clusters k.

The results of the full sweep are reported in Table 6, including the best parameter settings, normalized 3D positioning error $\tilde{\epsilon}_{3D_\beta}$ and normalized floor-hit $\tilde{\zeta}_\beta$ with the number of clusters $k = 8$ (CR = 7/3). We note that the best-case performance was chosen based on the lowest 3D positioning error ϵ_{3D} parameter. If the objective was to find the best floor-hit ζ, the chosen solution would differ in certain cases. Table 6 shows, that the parameters are unique for each individual dataset and that there is no universal parameter that ensures optimum positioning performance in every case. The aggregated results present the improve-

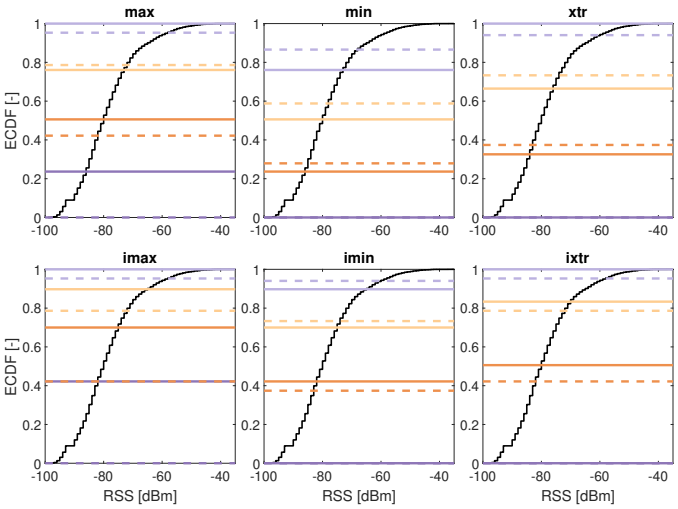

Fig. 8: Initial centroid values for 6 proposed k-means initializations (solid lines), and the corresponding centroid coordinates after clustering (dashed lines) performed on the dataset DSI 1 with k = 4.

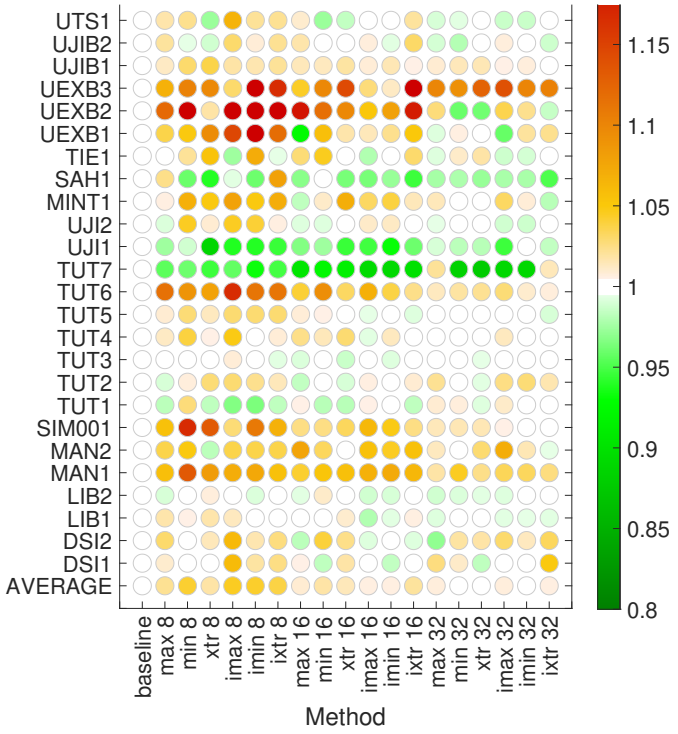

Fig. 9: Performance evaluation of the initialization schemes with k = 8, 16, and 32 across all datasets. The color of the dot refers to the normalized 3D positioning error $\tilde{\epsilon}_{3D_\alpha}$.

ment in positioning performance across both evaluation metrics. When compared to the results from Table 5, the parameter sweep found the parameters achieving 7% better normalized 3D positioning error $\tilde{\epsilon}_{3D_\beta}$ across all datasets (0.98 in Table 6 and 1.05 in Table 5).

For many datasets, the best-performing parameters are identical (TUT 6 or TUT 7) or very similar (DSI 1) to the β (Best Coef.) benchmark (see Table 6 and Table 3), if the k-means parameters are disregarded. Therefore, if

TABLE 5: Results based on β K-NN setting with max init. and Manhattan distance metric of k-means

	EWO8		EWO16		EWO32	
Dataset	$\tilde{\epsilon}_{3D\beta}$	$\tilde{\zeta}_\beta$	$\tilde{\epsilon}_{3D\beta}$	$\tilde{\zeta}_\beta$	$\tilde{\epsilon}_{3D\beta}$	$\tilde{\zeta}_\beta$
DSI 1	1.03	1.00	0.99	1.00	0.97	1.00
DSI 2	1.03	1.00	1.04	1.00	1.00	1.00
LIB 1	0.98	1.00	1.00	1.00	1.00	1.00
LIB 2	1.03	1.00	1.01	1.00	1.00	1.00
MAN 1	1.11	1.00	1.06	1.00	1.01	1.00
MAN 2	1.12	1.00	1.02	1.00	1.04	1.00
MINT 1	1.02	1.00	0.98	1.00	0.99	1.00
SAH 1	0.97	1.03	1.03	0.99	0.99	0.99
SIM 1	1.10	1.00	1.06	1.00	1.01	1.00
TIE 1	0.90	1.04	0.91	1.07	0.91	1.07
TUT 1	1.03	1.00	1.01	1.01	0.99	1.00
TUT 2	1.17	0.99	1.07	1.01	1.07	1.01
TUT 3	1.00	1.00	1.00	1.00	1.00	1.00
TUT 4	1.03	0.99	1.01	1.00	1.00	1.00
TUT 5	1.05	1.00	1.02	1.00	1.01	1.00
TUT 6	1.10	1.00	1.06	1.00	1.02	1.00
TUT 7	1.13	1.00	1.08	1.00	1.02	1.00
UEX B1	1.11	0.97	1.01	0.98	0.99	1.01
UEX B2	1.03	0.98	1.11	0.99	1.01	0.99
UEX B3	1.11	1.03	1.07	1.13	1.02	1.00
UJI 1	1.05	1.00	1.03	1.00	1.02	1.00
UJI 2	1.05	0.98	1.01	0.99	1.01	0.99
UJI B1	1.07	1.00	1.02	1.00	1.01	1.00
UJI B2	0.98	1.00	0.98	1.00	1.00	1.00
UTS 1	1.03	1.02	1.03	1.01	1.02	1.00
Average	1.05	1.00	1.02	1.01	1.01	1.00

TABLE 6: Best-case results from the full parameter sweep

		Config. k-means		Config. K-NN		EWO8	
Dataset	data rep.	Init.	distance	K	distance	$\tilde{\epsilon}_{3D\beta}$	$\tilde{\zeta}_\beta$
DSI 1	pow	max	Sq. Euclidean	8	Sørensen	0.93	1
DSI 2	pos	xtr	Sq. Euclidean	17	Sørensen	0.96	1
LIB 1	pos	xtr	Manhattan	14	Euclidean	0.97	1
LIB 2	pos	ixtr	Sq. Euclidean	10	Sørensen	1.01	1
MAN 1	exp	imax	Sq. Euclidean	23	Sørensen	1	1
MAN 2	pos	max	Manhattan	35	Euclidean	0.92	1
MINT 1	pow	ixtr	Manhattan	22	Euclidean	0.96	1
SAH 1	pow	xtr	Manhattan	35	Sørensen	0.84	1.12
SIM 1	exp	imin	Manhattan	31	Neyman	1.04	1
TIE 1	pos	ixtr	Manhattan	28	PLGD10	0.89	1.11
TUT 1	pos	xtr	Manhattan	3	PLGD40	1.02	1
TUT 2	pow	imax	Sq. Euclidean	2	Sørensen	1.04	1.04
TUT 3	pos	imin	Sq. Euclidean	2	Sørensen	0.99	0.98
TUT 4	pos	imax	Manhattan	3	PLGD10	1.02	0.99
TUT 5	pos	min	Manhattan	3	PLGD10	1.03	0.99
TUT 6	pos	imin	Sq. Euclidean	1	Sørensen	1.04	1
TUT 7	pos	xtr	Sq. Euclidean	1	Sørensen	1.05	1
UEX B1	pos	max	Manhattan	3	Euclidean	1.02	0.95
UEX B2	pos	max	Manhattan	4	Euclidean	0.98	0.98
UEX B3	exp	min	Sq. Euclidean	2	Neyman	0.99	1.13
UJI 1	pow	min	Sq. Euclidean	8	Sørensen	1.04	1
UJI 2	exp	imin	Manhattan	30	Neyman	0.96	1.01
UJI B1	exp	xtr	Sq. Euclidean	35	Euclidean	0.93	1
UJI B2	pos	xtr	Manhattan	34	LGD	0.89	1
UTS 1	exp	max	Sq. Euclidean	23	Neyman	0.98	1.01
Average						0.98	1.01

applying the proposed compression scheme to an existing and evaluated dataset, it is likely that the previously found best-case parameters will remain optimal after applying the compression as well.

The results presented above show, that EWOk can significantly reduce the size of the IPS's radio map, without

degrading the actual positioning performance of the dataset. If the appropriate parameter sweep is performed, the compression can further boost the positioning performance. Consequently, we can conclude that applying EWOk and optimizing the parameters of K-NN lead to improvements in positioning performance on top of preserving energy and resources within the positioning system achieved with the compression.

5.4 Multidimensional Compression with PCA

To demonstrate the possible co-existence of EWOk with other compression or clustering mechanisms from the existing literature [20], [22], we combine it with additional, feature-space-wise compression scheme, namely PCA (although any other dimensionality reduction technique can be applied in practical system). PCA is incorporated into the system as specified in Section 4.

We evaluate the positioning and compression performance of the following schemes. First, we combine the PCA with the α (Simple) configuration K-NN, further denoted PCA. Second, we combine the PCA with EWOk (k = 8, Manhattan distance), followed by the α configuration K-NN, denoted as $PCA + EWO8$. We normalize both solutions towards the α baseline. We consider α baseline since the regularized distance metrics included in the β baseline are incompatible with the PCA method. The common parameter of the PCA compression is Thr = 90, specifying the minimum total variance left in the training features.

Table 7 lists the results of the evaluation on all datasets. We note, that the CR of the PCA method is calculated as CR_{pca}, and the CR of the $PCA + EWO8$ method is calculated as CR_{tot} (see Sec. 4). Additionally, the $\Delta_{\epsilon_{3D}}$ and Δ_{ζ} parameters characterize the potential improvement of the positioning. The aggregated result shows that the average CR of the PCA method equals 10.95, the average CR of the combination of $PCA + EWO8$ is equal to 25.54, and that the PCA achieves 2% smaller positioning error while evaluating with the same parameters at Thr = 90. At the presented settings, we trade 2% higher positioning error for more than 2.3 times larger CR when considering $PCA + EWO8$.

In the next part of the evaluation, we report only the aggregated results [58] of the PCA and $PCA + EWOk$ schemes across all datasets. Now, we consider multiple values of k as well, denoted in the abbreviation accordingly. Table 8 lists the Thr, k, and aggregated normalized 3D positioning errors $\tilde{\epsilon}_{3D_\alpha}$ for the considered schemes, along with their dissimilarities $\Delta_{\epsilon_{3D}}$ and Δ_{ζ}.

The aggregated results in Table 8 show the configurable CR and the corresponding trade-off in terms of 3D positioning error. The table compares the performance of the $PCA + EWOk$ method at k = 8, 16, and 32 with the PCA setting at different Thr levels. The results show the increasing $\tilde{\epsilon}_{3D_\alpha}$ with the increasing CR as the general trend, with several exceptions. When comparing $PCA + EWOk$ to the PCA methods at the same Thr levels, the normalized 3D positioning errors are comparable. If, on the other hand, we compare their performance at the same CR levels, e.g. $PCA+EWO16$ at 90 Thr (and k = 16) and PCA at 80 Thr, both achieving approx. 20 CR, the difference in 3D positioning error is substantial. A similar occurrence is observed at the CR = approx. 10. Table 8 shows, that when combining EWOk and PCA compression, the achieved positioning results are better than when using the PCA compression only while considering the same CR.

In order to demonstrate the effectiveness of EWOk in combination with PCA compression, we show the dependency of CR on the normalized 3D positioning error $\tilde{\epsilon}_{3D_\alpha}$ in Fig. 10. The figure plots the aggregated results for the standalone PCA compression (denoted PCA), as well as the combination of PCA with EWOk with k = 8, 16, and 32 (denoted $PCA+EWO8$, $PCA+EWO16$, and $PCA+EWO32$) on varying Thr levels. In case the maximum allowed positioning error increase due to the compression is set to 35%, PCA

TABLE 7: Performance of PCA and $PCA + EWO8$

	PCA			PCA + EWO8				
Dataset	$\tilde{\epsilon}_{3D\alpha}$	$\tilde{\zeta}_{\alpha}$	CR_{pca}	$\tilde{\epsilon}_{3D\alpha}$	$\tilde{\zeta}_{\alpha}$	CR_{tot}	$\Delta_{\epsilon_{3D}}$	Δ_{ζ}
DSI 1	1.09	1	4.36	1.21	1	10.18	−0.12	0
DSI 2	1.11	1	4.36	1.09	1	10.18	0.02	0
LIB 1	0.99	1	19.33	1.03	1	45.11	−0.04	0
LIB 2	0.99	1.02	13.13	0.99	1.02	30.64	0.00	0
MAN 1	1.13	1	3.11	1.21	1	7.26	−0.07	0
MAN 2	1.34	1	14.00	1.30	1	32.67	0.03	0
MINT 1	1.05	1	2.20	1.13	1	5.13	−0.07	0
SAH 1	1.31	1.16	22.79	1.28	1.22	53.19	0.02	−0.05
SIM 1	1.11	1	1.60	0.95	1	3.73	0.15	0
TIE 1	1.13	0.03	18.03	1.33	0.27	42.07	−0.17	−7
TUT 1	0.99	0.98	12.36	0.98	1	28.84	0.01	−0.02
TUT 2	0.90	1.17	13.62	0.92	1.12	31.77	−0.03	0.05
TUT 3	1.01	1	15.50	1.02	1	36.17	−0.01	0
TUT 4	1.01	1	13.97	1.06	1	32.60	−0.05	0
TUT 5	1.10	1.05	21.26	1.14	1.06	49.61	−0.03	−0.01
TUT 6	1.41	1	17.62	1.77	1	41.12	−0.26	0
TUT 7	1.22	1	22.25	1.36	1	51.92	−0.11	0
UEX B1	2.23	0.68	15.00	2.08	0.42	35.00	0.07	0.38
UEX B2	1.85	0.77	10.00	1.47	0.72	23.33	0.20	0.06
UEX B3	1.22	0.91	3.33	1.24	0.77	7.78	−0.02	0.16
UJI 1	0.85	1.03	8.13	0.84	1.02	18.96	0.02	0
UJI 2	1.03	1.01	7.76	1.03	1.01	18.11	0	0
UJI B1	0.99	1	1.50	1.04	1	3.50	−0.05	0
UJI B2	1.08	1	1.29	1.10	1	3.02	−0.02	0
UTS 1	1	1.01	7.10	1.07	1	16.56	−0.07	0.01
Average	1.17	0.95	10.94	1.19	0.94	25.54	−0.02	−0.26

TABLE 8: Aggregated results for different PCA threshold Thr and varying number of clusters

		PCA+EWOk			PCA	
Thr	k	CR_{tot}	$\tilde{\epsilon}_{3D_\alpha}$	$\Delta_{\epsilon_{3D}}$	$\tilde{\epsilon}_{3D_\alpha}$	CR_{pca}
99	8	5.97	1.10	−0.06	1.04	2.56
	16	4.48	1.06	−0.02		
	32	3.58	1.05	−0.01		
95	8	14.50	1.09	−0.01	1.08	6.21
	16	10.87	1.08	0.00		
	32	8.70	1.10	−0.01		
90	8	25.54	1.19	−0.02	1.17	10.94
	16	19.15	1.14	0.01		
	32	15.32	1.16	0.00		
80	8	48.00	1.25	−0.01	1.23	20.57
	16	36.00	1.21	0.01		
	32	28.80	1.21	0.01		
50	8	144.01	1.66	−0.06	1.59	61.72
	16	108.01	1.54	0.01		
	32	86.41	1.51	0.03		

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2023.3277333

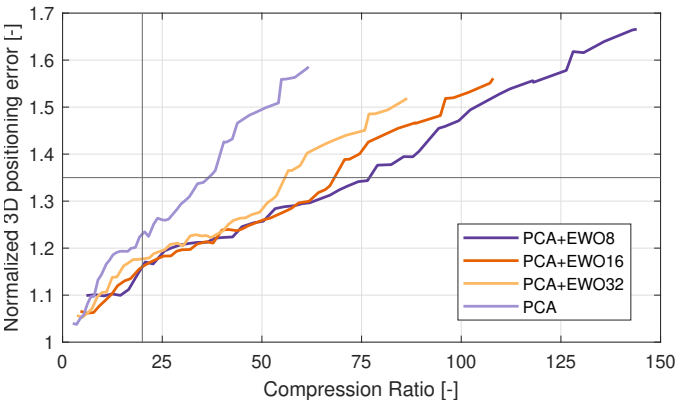

Fig. 10: Dependency of the CR on normalized 3D positioning error $\tilde{\epsilon}_{3D_\alpha}$ for the plain PCA and the proposed methods.

achieves 36.7 CR, while $PCA + EWO8$'s CR is 76.5 (see the horizontal line in Fig. 10). Alternatively, if the required CR is set to 20, $\tilde{\epsilon}_{3D_\alpha}$ increases by 23% when considering the PCA method, and 16% when considering either $PCA + EWO8$ or $PCA + EWO16$ solutions (see vertical line). The results unambiguously show the favorable trade-off between the compression efficiency and the positioning error of the proposed compression scheme combined with PCA, compared to the stand-alone PCA compression.

5.5 Discussion

In this section, we present the exhaustive evaluation of EWOk's impact on the positioning performance and its compression capabilities when applying it in the IPS. In the following lines, we summarize and discuss the most crucial findings and observations.

From evaluating the positioning performance when reducing the original 64-bit datasets to a common 7-bit data benchmark, we observe almost identical positioning accuracy. Similarly, the proposed EWOk further reduces the granularity of the individual values in the radio map and, in certain cases, results in improved positioning. This observation can be explained by high uncertainty in the data, which may be filtered out by reducing the data quality.

When utilizing EWOk on the indoor positioning dataset, it is highly recommended to consider the number of cluster k maximizing the compression efficiency, namely

$$k = 2^n \tag{12}$$

where $n = 1, 2...6$ to utilize the whole available alphabet of symbols in the compressed radio map. Namely $n = 3$ (8-means) offers a very good trade-off between the high CR and the tolerable positioning error.

The proposed non-random initialization schemes not only remove the uncertainties caused by the randomness and ensure advantageous positioning performance but at the same time reduce the number of iterations of the k-means, effectively reducing the algorithm's complexity.

In this work, we combine the proposed system with the PCA compression to demonstrate its compatibility. Nevertheless, it is possible to combine the EWOk compression scheme with numerous other methods proposed in the literature that could further increase the storage and processing speed efficiency, along with numerous other solutions that can co-exist with the proposed scheme including prior clustering to reduce the search-space of k-means [20], additional feature-space-based compression schemes [59], or any heuristic applied onto the dataset [60]. Similarly, the positioning algorithm can be freely chosen, not limited to K-NN or its alternatives.

6 Conclusion

This work proposes EWOk, an Element-Wise cOmpression using k-means, which reduces the radio map to up to 1% of its original size when combined with additional PCA feature-space dimensionality reduction. The proposed solution enables flexible adjustment of the CR, to obtain the desired storage and transfer savings while preserving high positioning performance using K-NN algorithm. The proposed positioning system is designed to be trained and validated on the network or cloud, and it aims to reduce the computational, memory, and data transfer load for the online prediction at the UE. We proposed the 7-bit data representation based on the current standardization as the benchmark for evaluating all RSS-based datasets and showed that using 7-bit representation does not degrade the data. The reported CRs achieved by the proposed method are substantially higher in case the benchmark is not based on the 7-bit data representation, as many datasets are represented as rational (floating point) numbers. In order to overcome the challenges related to the random initialization of the k-means algorithm, we proposed 6 non-random initialization methods derived from the input data distribution that ensure improved positioning performance and reduce the iterative process. We evaluate the proposed method on 25 RSS indoor positioning datasets in order to obtain impartial and unbiased results. The numerical results showed that EWOk compression achieves 2.3 fold radio map CR with only 5% positioning error increase on average, with a single iteration of the EWOk algorithm. In certain deployments, implementing the proposed compression scheme boosts the positioning performance in terms of 3D positioning error, as well as the floor-hit. Sweeping over the parameters can further boost the positioning performance significantly while preserving the valuable resources, as shown in Table 6. When combining EWOk with PCA, it is possible to reduce the complexity of K-NN and obtain many-fold higher CR with only a slight increase in 3D positioning error. The implementation is scalable (based on the dataset) to up to a 100-fold compression rate with a higher positioning error trade-off.

References

[1] S. He, W. Lin, and S.-H. G. Chan, "Indoor localization and automatic fingerprint update with altered ap signals," *IEEE Trans. on Mobile Computing*, vol. 16, no. 7, pp. 1897–1910, 2016.

[2] A. Ometov *et al.*, "A survey on wearable technology: History, state-of-the-art and current challenges," *Computer Networks*, vol. 193, p. 108 074, 2021.

[3] F. Zafari, A. Gkelias, and K. K. Leung, "A survey of indoor localization systems and technologies," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2568–2599, 2019.

[4] C. Laoudias *et al.*, "A survey of enabling technologies for network localization, tracking, and navigation," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3607–3644, 2018.

[5] J. Torres-Sospedra *et al.*, "A comprehensive and reproducible comparison of clustering and optimization rules in wi-fi fingerprinting," *IEEE Trans. on Mobile Computing*, 2020.

[6] L. Klus *et al.*, "Rss fingerprinting dataset size reduction using feature-wise adaptive k-means clustering," in *2020 12th International Congress on Ultra Modern Telecommunications and Control Systems*, 2020, pp. 195–200.

[7] C. D. Manning, *An introduction to information retrieval*. Cambridge university press, 2009.

[8] L. Klus *et al.*, "Towards accelerated localization performance across indoor positioning datasets," in *2022 International Conference on Localization and GNSS (ICL-GNSS)*, IEEE, 2022, pp. 1–7.

[9] R. F. Brena *et al.*, "Evolution of indoor positioning technologies: A survey," *Journal of Sensors*, vol. 2017, 2017.

[10] A. Nessa *et al.*, "A survey of machine learning for indoor positioning," *IEEE Access*, vol. 8, pp. 214 945–214 965, 2020.

[11] R. C. Luo and T.-J. Hsiao, "Indoor localization system based on hybrid wi-fi/ble and hierarchical topological fingerprinting approach," *IEEE Trans. on Vehicular Technology*, vol. 68, no. 11, pp. 10 791–10 806, 2019.

[12] S. Yoon *et al.*, "Acmi: Fm-based indoor localization via autonomous fingerprinting," *IEEE Trans. on Mobile Computing*, vol. 15, no. 6, pp. 1318–1332, 2015.

[13] C. Wu, Z. Yang, and Y. Liu, "Smartphones based crowdsourcing for indoor localization," *IEEE Transactions on Mobile Computing*, vol. 14, no. 2, pp. 444–457, 2014.

[14] S. He and S.-H. G. Chan, "Tilejunction: Mitigating signal noise for fingerprint-based indoor localization," *IEEE Trans. on Mobile Computing*, vol. 15, no. 6, pp. 1554–1568, 2015.

[15] P. Bahl and V. N. Padmanabhan, "Radar: An in-building rf-based user location and tracking system," in *Proceedings of IEEE INFOCOM 2000*, vol. 2, 2000, pp. 775–784.

[16] J. Jun *et al.*, "Low-overhead wifi fingerprinting," *IEEE Trans. on Mobile Computing*, vol. 17, no. 3, pp. 590–603, 2017.

[17] C. Wu, Z. Yang, and C. Xiao, "Automatic radio map adaptation for indoor localization using smartphones," *IEEE Transactions on Mobile Computing*, vol. 17, no. 3, pp. 517–528, 2017.

[18] C. Wu *et al.*, "Gain without pain: Accurate WiFi-based localization using fingerprint spatial gradient," *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies*, vol. 1, no. 2, pp. 1–19, 2017.

[19] J. Tan *et al.*, "Implicit multimodal crowdsourcing for joint rf and geomagnetic fingerprinting," *IEEE Trans. on Mobile Computing*, 2021.

[20] N. Marques, F. Meneses, and A. Moreira, "Combining similarity functions and majority rules for multi-building, multi-floor, wifi positioning," in *Int. Conf. on Indoor Positioning and Indoor Navigation*, IEEE, 2012, pp. 1–9.

[21] G. Caso, L. De Nardis, and M.-G. Di Benedetto, "A mixed approach to similarity metric selection in affinity propagation-based wifi fingerprinting indoor positioning," *Sensors*, vol. 15, no. 11, pp. 27 692–27 720, 2015.

[22] A. Razavi, M. Valkama, and E.-S. Lohan, "K-means fingerprint clustering for low-complexity floor estimation in indoor mobile localization," in *2015 IEEE Globecom Workshops*, 2015.

[23] J. Talvitie *et al.*, "Method and analysis of spectrally compressed radio images for mobile-centric indoor localization," *IEEE Trans. on Mobile Computing*, vol. 17, no. 4, pp. 845–858, 2017.

[24] S. Subedi and J.-Y. Pyun, "A survey of smartphone-based indoor positioning system using rf-based wireless technologies," *Sensors*, vol. 20, no. 24, 2020.

[25] P. Davidson and R. Piché, "A survey of selected indoor positioning methods for smartphones," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 1347–1370, 2016.

[26] B. Wang *et al.*, "A novel weighted knn algorithm based on rss similarity and position distance for wi-fi fingerprint positioning," *IEEE Access*, vol. 8, pp. 30 591–30 602, 2020.

[27] S. Liu, P. Zhu, and S. Qin, "An improved weighted knn algorithm for imbalanced data classification," in *2018 IEEE 4th International Conference on Computer and Communications (ICCC)*, IEEE, 2018, pp. 1814–1819.

[28] S. Zhang *et al.*, "Efficient knn classification with different numbers of nearest neighbors," *IEEE Trans. on neural networks and learning systems*, vol. 29, no. 5, pp. 1774–1785, 2017.

[29] J. Torres-Sospedra *et al.*, "Comprehensive analysis of distance and similarity measures for wi-fi fingerprinting indoor positioning systems," *Expert Systems with Applications*, vol. 42, no. 23, pp. 9263–9278, 2015.

[30] A. Cramariuc, H. Huttunen, and E. S. Lohan, "Clustering benefits in mobile-centric wifi positioning in multi-floor buildings," in *2016 International Conference on Localization and GNSS*, 2016.

[31] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[32] X. Zhao *et al.*, "Joint principal component and discriminant analysis for dimensionality reduction," *IEEE Trans. on neural networks and learning systems*, vol. 31, no. 2, pp. 433–444, 2019.

[33] Z. Xia, Y. Chen, and C. Xu, "Multiview pca: A methodology of feature extraction and dimension reduction for high-order data," *IEEE Trans. on Cybernetics*, 2021.

[34] A. Moreira, I. Silva, and J. Torres-Sospedra, "The dsi dataset for wi-fi fingerprinting using mobile devices, version 1.0," version 1.0, *Zenodo, Apr*, 2020.

[35] A. Moreira *et al.*, "Multiple simultaneous wi-fi measurements in fingerprinting indoor positioning," in *Int. Conf. on Indoor Positioning and Indoor Navigation*, 2017.

[36] J. Torres-Sospedra *et al.*, "UJIIndoorLoc: A new multi-building and multi-floor database for WLAN fingerprint-based indoor localization problems," in *Int. Conf. on Indoor Positioning and Indoor Navigation*, 2014, pp. 261–270.

[37] G. M. Mendoza-Silva *et al.*, "Ble rss measurements dataset for research on accurate indoor positioning," *Data*, vol. 4, no. 1, p. 12, 2019.

[38] G. M. Mendoza-Silva *et al.*, "Long-term wifi fingerprinting dataset for research on robust indoor positioning," *Data*, vol. 3, no. 1, 2018.

[39] F. J. Aranda *et al.*, "Multi-slot ble raw database for accurate positioning in mixed indoor/outdoor environments," *Data*, vol. 5, no. 3, p. 67, 2020.

[40] T. King *et al.*, *CRAWDAD dataset mannheim/compass (v. 2008-04-11)*, Downloaded from https : / / crawdad . org / mannheim / compass/20080411, Apr. 2008.

[41] T. King, T. Haenselmann, and W. Effelsberg, "On-demand fingerprint selection for 802.11-based positioning systems," in *2008 International Symposium on a World of Wireless, Mobile and Multimedia Networks*, Jun. 2008, pp. 1–8.

[42] X. Song *et al.*, "A novel convolutional neural network based indoor localization framework with wifi fingerprinting," *IEEE Access*, vol. 7, pp. 110 698–110 709, 2019.

[43] E.-S. Lohan *et al.*, "Wi-fi crowdsourced fingerprinting dataset for indoor positioning," *MDPI Data*, vol. 2, no. 4, 2017.

[44] P. Richter, E. S. Lohan, and J. Talvitie. "WLAN (WiFi) rss database for fingerprinting positioning." (Jan. 2018), [Online]. Available: https://zenodo.org/record/1161525.

[45] Lohan. "Additional TAU datasets for Wi-Fi fingerprinting- based positioning." version v1, 11.05.2020. (May 2020), [Online]. Available: https://doi.org/10.5281/zenodo.3819917.

[46] E. S. Lohan, J. Torres-Sospedra, and A. Gonzalez, *WiFi RSS measurements in Tampere University multi- building campus, 2017*, version 1, Zenodo, Aug. 2021.

[47] N. Saccomanno, A. Brunello, and A. Montanari, "What you sense is not where you are: On the relationships between fingerprints and spatial knowledge in indoor positioning," *IEEE Sensors Journal*, pp. 1–1, 2021.

[48] IEEE, "IEEE Standard for Information technology-Telecommunication and information exchange between systems-Local and metropolitan area networks-Specific requirements Part11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment1: Radio Resource Measurement of Wireless LANs," 2009.

[49] ETSI, "Wideband transmission systems; Data transmission equipment operating in the 2,4 GHz band; Harmonized Standard for access to radio spectrum," 2019.

[50] ETSI, "Wireless Access Systems (WAS); 5,8 GHz fixed broadband data transmitting systems; Harmonised Standard covering the essential requirements of article 3.2 of Directive 2014/53/EU," 2017.

[51] ETSI, "LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Requirements for support of radio resource management (3GPP TS 36.133 version 16.7.0 Release 16)," 2020.

[52] ETSI, "5G; NR; Requirements for support of radio resource management (3GPP TS 38.133 version 15.3.0 Release 15)," 2018.
[53] P. Fränti and S. Sieranoja, "How much can k-means be improved by using better initialization and repeats?" *Pattern Recognition*, vol. 93, pp. 95–112, 2019.
[54] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," Stanford, Tech. Rep., 2006.
[55] T. F. Gonzalez, "Clustering to minimize the maximum intercluster distance," *Theoretical computer science*, vol. 38, pp. 293–306, 1985.
[56] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *Advances in neural information processing systems*, vol. 14, 2001.
[57] J. B. Tenenbaum, V. d. Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *science*, vol. 290, no. 5500, pp. 2319–2323, 2000.
[58] J. Torres-Sospedra *et al.*, "Towards ubiquitous indoor positioning: Comparing systems across heterogeneous datasets," in *2021 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, IEEE, 2021, pp. 1–8.
[59] B. Jia *et al.*, "On the dimension reduction of radio maps with a supervised approach," in *2017 IEEE 42nd Conference on Local Computer Networks*, 2017, pp. 199–202.
[60] A. Abusara, M. S. Hassan, and M. H. Ismail, "Rss fingerprints dimensionality reduction in wlan-based indoor positioning," in *2016 Wireless Telecommunications Symposium (WTS)*, 2016.

Lucie Klus has been an Early Stage Researcher of A-WEAR project in joint Doctoral Degree programme at Tampere University, Finland, and Jaume I University, Spain, since September 2019. She received her master's degree in Electronics and Communications at Brno University of Technology in 2019. Her research interests include modern approaches in wireless communications, data analytics, optimization, and machine learning techniques. Her current research topic aims to improve the efficiency of data transfer and storage of wearable-originated data. She is a student member of IEEE.

Roman Klus is a Doctorate Researcher at Tampere University (TAU), Finland. He received his Ing. degree (Czech equivalent of master's) in the field of Electronics and Communications from Brno University of Technology in 2019. His research focuses on modern machine learning approaches in 5G and beyond networks, especially utilizing neural network structures in mobility management and positioning. He is a student member of IEEE.

Joaquín Torres-Sospedra is working at the ALGORITMI Research Centre (Universidade do Minho, Guimarães, Portugal) as an Postdoctoral fellow leading the MSCA-IF ORIENTATE. He has authored more than 120 articles in journals and conferences. His current research interests include indoor positioning solutions based on Wi-Fi & BLE, Machine Learning and Evaluation frameworks. He has supervised 5 Master and 2 PhD Students. He is the chair of the IPIN International Standards Committee and IPIN Smartphone-based off-site Competition.

E.S. Lohan is a Professor at Tampere University (TAU), Finland. She received an M.Sc. degree in electrical engineering from Polytechnics University of Bucharest, Romania, in 1997, a D.E.A. degree (French equivalent of master) in econometrics at Ecole Polytechnique, Paris, France, in 1998, and a Ph.D. degree in telecommunications from Tampere University of Technology in 2003. She is now a professor at the Electrical Engineering unit at Tampere University, Finland and the coordinator of the MSCA EU A-WEAR network. Her current research interests include wireless location techniques, wearable computing, and privacy-aware positioning solutions.

Carlos Granell (BS'98, MS'00, PhD'06) is an associate professor in Computer Science at the Universitat Jaume I of Castellón, Spain. He was previously a postdoctoral researcher at the European Commission–Joint Research Centre (Italy). His research interests lie in the multidisciplinary application of Geographic Information Science/Systems, spatial analysis and visualization of streams of sensor- and/or user-generated geographic content, and reproducibility research practices.

Jari Nurmi (S'87–M'95–SM'01) D.Sc.(Tech), works as a Professor at Tampere University, TAU (formerly Tampere University of Technology, TUT), Finland since 1999, in the Electrical Engineering unit. He is working on embedded computing systems, wireless localization, and software-defined radio and -networks. He held various research, education and management positions at TUT since 1987 and was the Vice President of VLSI Solution Oy 1995-1998. He has edited five Springer books, and has published over 350 international conference and journal articles and book chapters. He is the director of national DELTA doctoral training network of over 200 PhD students, coordinator of the European doctoral training network APROPOS, and the head of A-WEAR European joint PhD degree program at TAU.