A Weighted Kernel-based Hierarchical Classification Method for Zoning of Sensors in Indoor Wireless Networks

Daniel Alshamaa, Farah Mourad-Chehade
Institut Charles Delaunay, ROSAS, LM2S
Université de Technologie de Troyes
UMR 6281, CNRS, Troyes, France
daniel.alshamaa@utt.fr, farah.chehade@utt.fr

Paul Honeine
LITIS lab
Université de Rouen Normandie
Rouen, France
paul.honeine@univ-rouen.fr

***Abstract*—This paper presents a solution for localization of sensors by zoning, in indoor wireless networks. The problem is tackled by a classification technique, where the objective is to classify the zone of the mobile sensor for any observation. The method is hierarchical and uses the belief functions theory to assign confidence levels for zones. For this purpose, kernel density estimation is used first to model the features observations. The algorithm then uses hierarchical clustering and similarity divergence, creating a two-level hierarchy, to reduce the number of zones to be classified at a time. At each level of the hierarchy, a feature selection technique is carried to optimize the misclassification rate and feature redundancy. Experiments are realized in a wireless sensor network to evaluate the performance of the proposed method.**

***Index Terms*—Belief functions, classification, feature selection, hierarchical clustering, kernel density estimation, zoning.**

I. Introduction

Localization of sensors is a key aspect in wireless networks, as the knowledge of the sensor's position is essential to process the retrieved information [1]. This paper considers a zoning approach, where the objective is to determine the zone where the mobile sensor resides, rather than the exact position. This issue is important for the health-care domain for instance, where Alzheimer's patients might be lost in their nursing home [2], in museums for supporting guides and emergency management [3], for large malls to facilitate shopping [4], etc and where locating people in a specific zone of such environments is completely sufficient. The most widely adopted approach in indoor localization is wireless fingerprinting [5]. Wireless fingerprinting leverages the available wireless transceivers along with the already deployed networking infrastructure. This approach requires an offline training phase and an online localization phase. The problem is then formulated as multi-class classification, where the aim is to classify the zone of the mobile sensor according to the measured observation.

In this paragraph, we provide a succinct survey of the classification methods that exist in literature. Reasearchers have proposed techniques that are based on the concept of a perceptron, where a sum of weighted inputs is computed and the output is compared to a threshold in order to choose a class [6]. However, perceptrons work only for instances that are linearly separable. When this is not the case, artificial neural networks were developed to solve the problem [7]. Another well-known nonlinear method is Support Vector Machines (SVM) that classifies the instances using a decision surface or hyperplane that maximizes the margin between the classes [8]. The k-nearest neighbors algorithm determines the class of an instance by examining the labels of its nearest neighbors and voting for the most frequent one [9]. In addition, naive Bayes classifiers assume independency between features to release probabilistic output [10]. Logistic regression fits data into a logistic function and distributes probabilities on classes according to the generated function [11]. In another category, hierarchical approaches have been also proposed. Random forests [12] is an ensemble of trees, obtained by bootstrap sampling and by randomly changing the feature set during learning. More precisely, at each node in the decision tree, a random subset of the input attributes is taken, and the best feature is selected from this subset instead of all attributes. Hierarchical methods could also be derived from classical techniques. HSVM [13] solves a series of max-cut problems to recursively partition the classes into two-subsets, till pure leaf nodes having only one class are obtained. Then, the classical SVM is applied to solve the binary classification problem at each internal node. In our previous work [14], we proposed a classification technique that creates a two-level hierarchy using divergence-based clustering. The technique applies feature selection and the belief functions theory (BFT) to assign confidence levels for classes using the observations distributions.

This paper extends the classification method proposed in [14]. One major drawback of the previous method is that it requires a parametric distribution for data fitting. This, however, might not be the case in many practical classification problems, where the data fail to fit into one of the parametric distributions with an acceptable significance level. This paper proposes a kernel-based approach for the developed classification technique. The contributions of this paper are the following. The kernel density estimation to model the data observations is studied first. Afterwards, the hierarchical strategy using the new constructed model is investigated. An extension of the feature selection technique to consider both misclassification rate and feature redundancy is also presented. Finally, the proposed technique is used to localize the sensors by classifying the zones of the targeted area, according to an observation it receives from the surrounding Access Points.

The remainder of the paper is organized as follows. Section II describes the classification technique. Section III demonstrates the application in wireless networks for zoning localization of sensors. Section IV concludes this paper.

II. Classification Method

A. Problem statement

The classification problem is formulated as follows. Let

- $y_j^{cl}, j \in \{1, \ldots, m\}$ be the m competing classes;
- $F = \{f_1, \ldots, f_p\}$ be the set of p features;
- $\mathbf{x}_{j,r} = (x_{j,1,r}, \ldots, x_{j,p,r}), r \in \{1, \ldots, \ell_j\}$ be ℓ_j offline training observations labeled in y_j^{cl}, with respect to F;
- $\bar{\mathbf{x}} = (\bar{x}_1, \ldots, \bar{x}_p)$ be a new observation measured with respect to features F, such that $\bar{x}_k$ is its k-th element.

The aim of the algorithm is to find a function $\mathbf{h} : \mathbb{R}^p \rightarrow [0,1]^m$ such that $\mathbf{h}(\bar{\mathbf{x}}) = (\mathcal{C}f(y_1^{cl}), ..., \mathcal{C}f(y_m^{cl}))$, where $\mathcal{C}f(y_j^{cl})$ is the level of confidence of the statement: "$\bar{\mathbf{x}}$ *belongs to class* y_j^{cl}", for any new observation $\bar{\mathbf{x}} \in \mathbb{R}^p$. To do this, a clustering algorithm is first proposed to dispatch the classes within clusters denoted $y_i^C,\ i \in \{1, ..., N_C\}$. The classification problem consists then in finding the cluster at which belongs a measured observation and then to find its class within the selected cluster.

B. Kernel-based model

The proposed method creates a model that represents the distribution of the offline training observations of each class, so that once a new observation is measured online, the best matching model is the most probable to which the observation belongs. One approach consists in modelling parametrically the data, by fitting them into one of the known parametric distributions as proposed in [14]. However, when the assumptions of a parametric model fail, a more general non-parametric approach is required to estimate the probability density function of the measurements [15]. A solution is to construct a histogram of the data [16]. However, this depends on the starting position of the bins and their number, and suffers from the curse of dimensionality as the number of bins grows exponentially with dimensions, thus making this solution unsuitable for most applications [15]. For these reasons, kernel density estimation (KDE) is proposed in the following to model the training observations.

Suppose a region $\mathcal{S}_j$ is a hypercube that encloses o_j observations with side length h and centered at an estimation point $\tilde{\mathbf{x}} = (\tilde{x}_k), k \in \{1, \ldots, p\}$. We study the uni-variate case first. To find the number o_j of observations falling within $\mathcal{S}_j$, we consider the indicator function $I(u)$ defined such that,

$$I(u) = \begin{cases} 1, & \text{if } |u| < \frac{1}{2}; \\ 0, & \text{elsewhere.} \end{cases} \tag{1}$$

This function is known as a Parzen window or naive estimator. The quantity $I(\frac{\tilde{x}_k - x_{j,k,r}}{h})$ is then equal to unity if $x_{j,k,r}$ is inside $\mathcal{S}_j$ or 0 otherwise. The number of measurements within $\mathcal{S}_j$ is then computed as,

$$o_j = \sum_{r=1}^{\ell_j} I\left(\frac{\tilde{x}_k - x_{j,k,r}}{h}\right). \tag{2}$$

The density estimate $Q_{KDE,j}(\tilde{x}_k)$ is calculated as,

$$Q_{KDE,j}(\tilde{x}_k) = \frac{o_j}{\ell_j \times h}. \tag{3}$$

Then, by substituting Eq. (2) in Eq. (3), we obtain,

$$Q_{KDE,j}(\tilde{x}_k) = \frac{1}{\ell_j \times h} \sum_{r=1}^{\ell_j} I\left(\frac{\tilde{x}_k - x_{j,k,r}}{h}\right). \tag{4}$$

This model ensures that the observations close to $\tilde{x}_k$ contribute more than the far ones. However, the resulting density is bumpy, yielding discontinuous density estimates. Instead of assigning equal weights to all neighboring observations, the Parzen window is replaced by a smoother kernel $\mathcal{K}(u)$, such as triangular, Epanechnikov, cosine, logistic, Gaussian kernels, etc. The kernel density function is then given by,

$$Q_{KDE,j}(\tilde{x}_k) = \frac{1}{\ell_j \times h} \sum_{r=1}^{\ell_j} \mathcal{K}\left(\frac{\tilde{x}_k - x_{j,k,r}}{h}\right). \tag{5}$$

Since the shape of the kernel has a small effect on the model [17], a Gaussian kernel is considered due to the facility of its analytical derivations,

$$\mathcal{K}(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}u^2}. \tag{6}$$

The problem is to determine the bandwidth, or the smoothing parameter h. A small value of h overfits the data and makes it hard to interpret, while a large value over-smooths the KDE and masks the structure of the data. A practical approach to estimate h, is to maximize the pseudo-likelihood leave-one-out cross-validation. The bandwidth h is computed by maximizing,

$$ML(h) = \ell_j^{-1} \sum_{r=1}^{\ell_j} \log\left[\sum_{r' \neq r} \mathcal{K}\left(\frac{x_{j,k,r'} - x_{j,k,r}}{h}\right)\right] - \log[(\ell_j - 1)h]. \tag{7}$$

The KDE is easily extended to the multivariate case,

$$Q_{KDE,j}(\tilde{\mathbf{x}}) = \frac{1}{\ell_j \times h^p} \sum_{r=1}^{\ell_j} \mathcal{K}\left(\frac{\tilde{\mathbf{x}} - \mathbf{x}_{j,r}}{h}\right). \tag{8}$$

However, the same bandwidth is taken here on all axes, weighting all features equally. A good alternative for multivariate KDE is the product kernel,

$$Q_{KDE,j}(\tilde{\mathbf{x}}) = \frac{1}{\ell_j} \sum_{r=1}^{\ell_j} \frac{1}{h_1 \ldots h_p} \prod_{k=1}^{p} \mathcal{K}\left(\frac{\tilde{x}_k - x_{j,k,r}}{h_k}\right), \tag{9}$$

such that the bandwidth h_k is calculated at each feature dimension. It is worth noting here that by considering the product of kernels, we use only kernel independence which does not imply that we assume features independence.

C. Clustering Algorithm

The classification method proposed in [14] creates a two-level hierarchy, the first being a set of clusters, and the second being a set of classes of each cluster. The hierarchy is built based on the Kullback-Leibler divergence (D_{KL}) between parametric distributions representing the classes or clusters.

However, since a kernel density estimation is adopted here, the calculation of the D_{KL} is not trivial.

The D_{KL} between the two functions $Q_{KDE,j}(\mathbf{x})$ and $Q_{KDE,j'}(\mathbf{x})$ with respective density functions q_j and $q_{j'}$ of input $\mathbf{x}$ is defined as [18],

$$D_{KL}(Q_{KDE,j}||Q_{KDE,j'}) = \int_{\mathbf{x}} \log\left(\frac{q_j(\mathbf{x})}{q_{j'}(\mathbf{x})}\right) q_j(\mathbf{x}) d\mathbf{x}. \quad (10)$$

The expected value of the function $\log\left(\frac{q_j(\mathbf{x})}{q_{j'}(\mathbf{x})}\right)$ with respect to $q_j(\mathbf{x})$ is given by,

$$E_{q_j(\mathbf{x})}\left[\log\left(\frac{q_j(\mathbf{x})}{q_{j'}(\mathbf{x})}\right)\right] = \int_{\mathbf{x}} \log\left(\frac{q_j(\mathbf{x})}{q_{j'}(\mathbf{x})}\right) q_j(\mathbf{x}) d\mathbf{x}. \quad (11)$$

As recommended in [19], the expected value is approximated by

$$E_{q_j(\mathbf{x})}\left[\log\left(\frac{q_j(\mathbf{x})}{q_{j'}(\mathbf{x})}\right)\right] = \frac{1}{\ell_j}\sum_{r=1}^{\ell_j} \log\left(\frac{q_j(\mathbf{x}_{j,r})}{q_{j'}(\mathbf{x}_{j,r})}\right). \quad (12)$$

The D_{KL} is then deduced using Eqs. (11) and (12),

$$\begin{aligned} D_{KL}(Q_{KDE,j}||Q_{KDE,j'}) &= E_{q_j(\mathbf{x})}\left[\log\frac{q_j(\mathbf{x}_{j,r})}{q_{j'}(\mathbf{x}_{j',r})}\right] \quad (13) \\ &= \frac{1}{\ell_j}\sum_{r=1}^{\ell_j} \log\frac{q_j(\mathbf{x}_{j,r})}{q_{j'}(\mathbf{x}_{j',r})} \\ &= \frac{1}{\ell_j}\sum_{r=1}^{\ell_j} \log q_j(\mathbf{x}_{j,r}) - \log q_{j'}(\mathbf{x}_{j',r}). \end{aligned}$$

It is then easy to compute the D_{KL} by replacing the functions q_j and $q_{j'}$ by their kernel models obtained in the previous paragraph. Once the D_{KL} is calculated, the described clustering algorithm in [14] can be executed. At the end of the algorithm, a two-level hierarchy is created, the first level being a set of N_C optimal clusters $y_i^C, i \in \{1, \ldots, N_C\}$, and the second being a set of classes $y_j^{cl}, j \in J_i$ of each cluster y_i^C, such that J_i is the set of indices of classes y_j^{cl} in y_i^C. The objective becomes to classify clusters at first, then classes within clusters afterwards.

D. Feature selection technique

The objective of feature selection is to select the best subset of features according to a certain criteria. In [14], we proposed a technique that considers only the misclassification rate. However, another important factor to study is redundancy between features [20]. The selected features must not only be highly discriminant to reduce the misclassification rate, but also not redundant between each other. Indeed, having redundant features means that the same information is carried by these features, and thus similar classification accuracy can be obtained with fewer ones [21]. The selection must fulfill the two objectives; minimization of the misclassification rate, and minimization of the redundancy between the features. To do this, all the nonempty subsets F' of F are considered. Let $F' \subseteq F$ denote one considered subset. Since the feature selection technique will be applied at the clusters and the classes levels, unique notations for both are considered in the following, that is, let y denote either a class or a cluster, and K be their numbers.

The first objective, the error rate of subset F', is defined in [14] as follows,

$$\mathcal{E}(F') = 2^{-DisC(F')}. \quad (14)$$

such that $DisC(F')$, the discriminative capacity of the subset of features F', is computed as follows,

$$DisC(F') = \sum_{u=1}^{K}\sum_{v=1}^{K} D_{KL}(Q_{F',u}||Q_{F',v}), \quad (15)$$

$D_{KL}(Q_{F',u}||Q_{F',v})$ being the Kullback-Leibler divergence measured between kernel density estimations of entities u and v, with respect to F'.

To take into account the features redundancy, we consider the coefficient of multiple correlation R. This coefficient measures the level of dependency of a feature upon other ones. The square of the coefficient of multiple correlation R^2 of f_k in F' with respect to $F' \setminus \{f_k\}$ is defined as,

$$R_k^2 = c_k^T R_{xx,k}^{-1} c_k, \quad (16)$$

where c_k is the column vector with entries $r_{f_{k'}f_k}$ for $f_{k'} \in F' \setminus \{f_k\}$, $r_{f_{k'}f_k}$ being the correlation between $f_{k'}$ and f_k, and $R_{xx,k}^{-1}$ the inverse of the matrix of entries $r_{f_{k'}f_{k''}}$ for $f_{k'}$ and $f_{k''} \in F' \setminus \{f_k\}$. The redundancy between all the features of F' is the average of the multiple correlation coefficients of all $f_k \in F'$,

$$\mathcal{R}(F') = \sum_k \frac{R_k}{|F'|}, \quad (17)$$

where $|F'|$ is the cardinal of F'.

The feature selection technique searches for $F_s \subseteq F$ that minimizes both $\mathcal{E}(F_s)$ and $\mathcal{R}(F_s)$. We consider a search algorithm with a backward elimination strategy. The algorithm starts with the complete set of features F and eliminates continuously a feature, whose elimination satisfies the two objective functions. Let F_a be the features subset chosen at iteration $a \geq 1$, with $F_0 = F$ and the cardinal $|F_a|$ of F_a equal to $p - a$. At each iteration $a \geq 1$, all the subsets of F_{a-1} having $p - a$ elements are considered. Let $F_a^{(\eta)}$, $\eta = 1, ..., p - a + 1$, denote these subsets. We define the function $g_a\big(F_a^{(\eta)}\big)$ as follows,

$$g_a(F_a^{(\eta)}) = \alpha\frac{\mathcal{E}(F_{a-1}) - \mathcal{E}(F_a^{(\eta)})}{\max(\mathcal{E}(F_{a-1}), \mathcal{E}(F_a^{(\eta)}))} + (1-\alpha)\frac{\mathcal{R}(F_{a-1}) - \mathcal{R}(F_a^{(\eta)})}{\max(\mathcal{R}(F_{a-1}), \mathcal{R}(F_a^{(\eta)}))}, \quad (18)$$

where $\alpha \in [0, 1]$ is a tradeoff parameter chosen by the user to assign a weight for each objective. A positive value of $g_a\big(F_a^{(\eta)}\big)$ means that the subset $F_a^{(\eta)}$ is better than F_{a-1} in optimizing the objectives. The greater $g_a(\cdot)$ is, the better the subset is. This results in a selected subset at iteration a, $F_a = \arg\max_\eta g_a\big(F_a^{(\eta)}\big)$. A negative value of g indicates that no significant improvement is obtained in the objectives for the considered parameters and hence iterations stop when all $g_a\big(F_a^{(\eta)}\big)$, $\eta = 1, ..., p - a + 1$, are negative and the set of features $F_s = F_{a-1}$ is thus chosen.

Applying this technique at each level of the two-level hierarchy leads on one hand, to an optimal subset of features that is best to distinguish between the clusters, and on the other hand, to subsets of features that should be used for classification between the classes within each cluster.

E. Weighted decision using belief functions

Let y, denoting a cluster y^C or a class y^{cl} within a cluster, be a discrete variable taking values in $Y = \{y_1, \ldots, y_K\}$ and let 2^Y be the set of all the supersets of Y, i.e, $2^Y = \{\emptyset, \{y_1\}, \ldots, Y\}$. A fundamental function of the BFT is the mass function, also called the basic belief assignment (BBA). The mass $m(A)$ given to $A \in 2^Y$ stands for the proportion of evidence, brought by the source F_s, saying that the observed variable belongs to A. A detailed work on the implementation of belief functions for classification is found in [14]. To define the features BBAs, all observations related to the selected subset of features belonging to a set $A \in 2^Y$ are represented by kernel density estimation, $Q_{KDE,A}$. Then, having a new observation $\bar{\mathbf{x}}$, the mass $m(A)$ is calculated as follows,

$$m(A) = \frac{Q_{KDE,A}(\bar{\mathbf{x}})}{\sum_{A' \in 2^Y,} Q_{KDE,A'}(\bar{\mathbf{x}})}, \quad A \in 2^Y. \tag{19}$$

For decision making, the BFT uses the pignistic transformation [22]. It is defined as follows,

$$BetP(A) = \sum_{A \subseteq A'} \frac{m(A')}{|A'|}, \tag{20}$$

where A is a singleton of 2^Y. This equation is applied at the clusters level and the classes level within each cluster, leading respectively to $BetP^C(\{y_i^C\})$, $i \in \{1, ..., N_C\}$, and $BetP^{cl,i}(\{y_j^{cl}\})$, $j \in J_i$. To assign confidence levels for the classes, the pignistic levels of classes and clusters are combined as follows,

$$\mathcal{C}f(y_j^{cl}) = BetP^C(\{y_i^C\}) \times BetP^{cl,i}(\{y_j^{cl}\}), \tag{21}$$

such that $j \in J_i, i \in \{1, ..., N_C\}$. The class $\hat{y}_j^{cl}$ having the highest confidence is then selected,

$$\hat{y}_j^{cl} = \arg \max_{j \in J_i, i \in \{1,\ldots,m\}} \mathcal{C}f(y_j^{cl}). \tag{22}$$

III. Zoning of Sensors in Wireless Networks

The proposed classification technique is applied in indoor wireless networks for zoning of sensors. The objective is to determine the zone where the mobile sensor resides according to an observation measurement received from its environment. The mobile sensor moves in the targeted area and collects received signal strength indicators (RSSIs) from the WiFi Access Points (APs) installed in the network. The proposed method is used to classify the zone of the mobile sensor for a new measurement. In this case, the zones, the APs, and the RSSIs resemble the classes, the features, and the observations respectively. Experiments are conducted in the statistical and operational research department of the University of Technology of Troyes, France. The considered floor has a section area of 500 m^2 partitioned into 18 zones. It is noted that 23 AP networks were detected in the targeted area. Here, it is distinguished between the physical AP and the AP network, as for the same physical AP, there exist several networks. These networks give access to various populations (visitors, residents, staff), and use different channel bands (2.4 GHz, 5 GHz). Since the parameters of each network are controlled by the IT service as required, these networks are considered to be multiple features, although they carry some information redundancy. It is the role of the proposed feature selection technique to choose the best discriminating subset of APs, that is less redundant too. In each zone, a set of 30 measurements is considered to construct the databases and train the classifiers. Another set of 20 new observations in each zone is taken in another day to test the proposed method, as measurements of the same day are strongly dependent.

TABLE I: Influence of parameter α on overall accuracy and processing time.

Parameter α	number of selected APs	accuracy (%)	online time (s)
-	23	89.44	0.3168
0.25	8	83.89	0.2117
0.5	13	86.11	0.2619
0.75	18	92.78	0.2955

The collected RSSIs are represented by the KDE using a Gaussian kernel, and a bandwidth h calculated by maximizing the likelihood cross validation. The two-level hierarchy is then constructed by creating a dendrogram based on Kullback-Leibler divergence, then cutting it by maximizing the inter- and intra- cluster scatters, to obtain $N_C = 7$ optimal clusters. The feature selection technique is applied afterwards at each level of the hierarchy, by varying the parameter α to obtain the best subset of features according to misclassification rate and redundancy. The BFT is used to assign confidence levels to each zone based on the constructed hierarchy and the selected APs. Fig. 1 shows the importance of the KDE in modeling of the observations as compared to the parametric fitting, when the data do not really follow a parametric distribution. It shows also that the kernel shape does not have a significant impact on the model. Table I studies the influence of the user-defined parameter α on the number of selected APs, and on the performance of the method. As compared to the first line where no feature selection is carried, this technique can add an accuracy of 3% for $\alpha = 0.75$. A gain in the online time is also noted. Table II compares the performance of the proposed method with other conventional classification techniques. As the table shows, this proposed method carries an additional 5% of accuracy as compared to the previous technique, which also consists of a feature selection technique, outperforming all other described methods in terms of overall accuracy. Regarding the processing time, the proposed algorithm takes longer time to be executed due to the maximum likelihood and the following clustering phase. A gain is however noted by optimizing the set of features, reducing the dimensionality by neglecting redundant features, and thus reducing complexity.

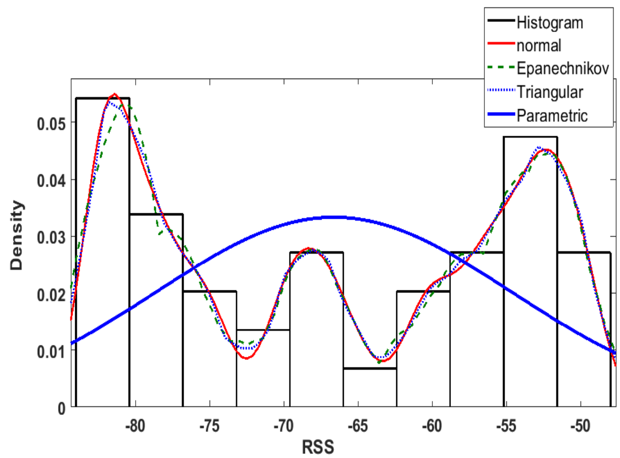

Fig. 1: Fitting of parametric normal distribution, and KDE of Gaussian, Epanechnikov, Triangular kernels, and of $h = 1.6$, of histogram of RSSI.

TABLE II: Comparison between classification techniques for zoning of sensors in indoor wireless networks, in terms of overall accuracy (%) and online processing time (s).

Technique	accuracy (%)	online time (s)
K-nearest neighbors [9]	83.33	0.1289
Naive Bayes [10]	81.66	0.1018
Multinomial logistic regression [11]	82.78	0.1498
Neural networks [7]	84.72	0.1866
Support Vector Machines [8]	85.55	0.1859
Random forests [12]	86.66	0.4077
Hierarchical Support Vector Machines [13]	86.38	0.4667
Previous method [14]	87.77	0.2508
Proposed method	92.78	0.2955

IV. Conclusion and Future Work

This paper tackled the problem of zoning localization of sensors by a classification technique. The proposed method extended a previous technique to the non-parametric case by kernel density estimation modeling. Moreover, the feature selection technique is extended to optimize both misclassification rate and feature redundancy. Experiments in an indoor environment to localize sensors prove the advantage carried by the proposed method in terms of overall accuracy, and its competence as compared to other state-of-the-art techniques. Future work will focus on auto-tuning the parameter α, and investigating an optimal number of hierarchy levels.

Acknowledgment

The authors would like to thank the European Regional Development Fund and the Grand Est Region in France for funding this work.

References

[1] D. Alshamaa, F. Mourad-Chehade, and P. Honeine, "Zoning-based localization in indoor sensor networks using belief functions theory," in *Signal Processing Advances in Wireless Communications (SPAWC), 2016 IEEE 17th International Workshop on*, pp. 1–5, IEEE, 2016.

[2] Y.-C. Liu, Y.-K. Ou, S.-N. Lin, and C.-W. Fang, "A study of the indoor walking navigation system for patients with early-stage Alzheimers disease," in *International Conference on Computer, Networks and Communication Engineering (ICCNCE 2013)*, Atlantis Press, 2013.

[3] A. Smirnov, N. Shilov, and A. Kashevnik, "Ontology based mobile smart museums service," in *AFIN 2012. The Fourth International Conference on Advances in Future Internet*, pp. 48–54, Citeseer, 2012.

[4] S. Wang, S. Fidler, and R. Urtasun, "Lost shopping! monocular localization in large indoor spaces," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2695–2703, 2015.

[5] D. Alshamaa, F. Mourad-Chehade, and P. Honeine, "Tracking of mobile sensors using belief functions in indoor wireless networks," *IEEE Sensors Journal*, vol. 18, no. 1, pp. 310–319, 2017.

[6] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," *Emerging artificial intelligence applications in computer engineering*, vol. 160, pp. 3–24, 2007.

[7] R. Rojas, *Neural networks: a systematic introduction*. Springer Science & Business Media, 2013.

[8] P. Honeine, Z. Noumir, and C. Richard, "Multiclass classification machines with the complexity of a single binary classifier," *Signal Processing*, vol. 93, no. 5, pp. pp. 1013–1026, 2013.

[9] R. Souza, L. Rittner, and R. Lotufo, "A comparison between k-optimum path forest and k-nearest neighbors supervised classifiers," *Pattern recognition letters*, vol. 39, pp. 2–10, 2014.

[10] V. Narayanan, I. Arora, and A. Bhatia, "Fast and accurate sentiment classification using an enhanced naive Bayes model," in *International Conference on Intelligent Data Engineering and Automated Learning*, pp. 194–201, Springer, 2013.

[11] D. Liu, T. Li, and D. Liang, "Incorporating logistic regression to decision-theoretic rough sets for classifications," *International Journal of Approximate Reasoning*, vol. 55, no. 1, pp. 197–210, 2014.

[12] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[13] Y. Chen, M. M. Crawford, and J. Ghosh, "Integrating support vector machines in a hierarchical output space decomposition framework," in *Geoscience and Remote Sensing Symposium (IGARSS), 2004 IEEE International*, vol. 2, pp. 949–952, IEEE, 2004.

[14] D. Alshamaa, F. M. Chehade, and P. Honeine, "A hierarchical classification method using belief functions," *Signal Processing*, vol. 148, pp. 68 – 77, 2018.

[15] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance," *Proceedings of the IEEE*, vol. 90, no. 7, pp. 1151–1163, 2002.

[16] V. Moghtadaiee, A. G. Dempster, and S. Lim, "Indoor localization using FM radio signals: A fingerprinting approach," in *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on*, pp. 1–7, IEEE, 2011.

[17] A. Eckert-Gallup and N. Martin, "Kernel density estimation (kde) with adaptive bandwidth selection for environmental contours of extreme sea states," in *OCEANS 2016 MTS/IEEE Monterey*, pp. 1–5, IEEE, 2016.

[18] J. Harmouche, C. Delpha, and D. Diallo, "Incipient fault amplitude estimation using KL divergence with a probabilistic approach," *Signal Processing*, vol. 120, pp. pp. 1–7, 2016.

[19] E. C. Anderson, E. G. Williamson, and E. A. Thompson, "Monte carlo evaluation of the likelihood for ne from temporally spaced samples," *Genetics*, vol. 156, no. 4, pp. 2109–2118, 2000.

[20] S. Tabakhi and P. Moradi, "Relevance–redundancy feature selection based on ant colony optimization," *Pattern recognition*, vol. 48, no. 9, pp. 2798–2811, 2015.

[21] M. Osl, S. Dreiseitl, F. Cerqueira, M. Netzer, B. Pfeifer, and C. Baumgartner, "Demoting redundant features to improve the discriminatory ability in cancer data," *Journal of biomedical informatics*, vol. 42, no. 4, pp. 721–725, 2009.

[22] P. Smets, "Belief functions: the disjunctive rule of combination and the generalized Bayesian theorem," *International Journal of approximate reasoning*, vol. 9, no. 1, pp. 1–35, 1993.