İç Mekanda Kullanıcı Lokalizasyonu için Farklı Makine Öğrenmesi Algoritmalarının Çoğulluk Kuralı ile Birleştirilmesi

User Localization in an Indoor Environment by Combining Different Algorithms through Plurality Rule

Muzaffer Cem Ateş[1], Osman Emre Gümüşoğlu[1], Aslınur Çolak[1], Nilgün Fescioglu-Unver[1]
[1]Department of Industrial Engineering, TOBB University of Economics and Technology, Turkey
mates@etu.edu.tr, ogumusoglu@etu.edu.tr, a.colak@etu.edu.tr, nfunver@etu.edu.tr

***Özetçe*— İç mekan ortamında kullanıcı lokalizasyonu fabrikalar, akıllı evler, hastaneler, huzurevleri gibi üretim ve hizmet sistemlerini içeren geniş bir uygulama alanına sahiptir. Wi-Fi sinyallerine dayalı kullanıcı lokalizasyonu, çeşitli sınıflandırma algoritmaları kullanılarak geniş çapta incelenmiştir. Bu tür bir problemde, bir iç mekâna yerleştirilen birkaç Wi-Fi yönlendiricisi, kullanıcının konumuna bağlı olarak farklı güçlere sahip sinyaller sağlar. Çoğu sınıflandırma algoritması, kullanıcının konumunu yüksek doğruluk oranıyla tespit etmektedir. Bununla birlikte, mevcut literatürde bu sorunu çözmek için yaygın olarak kabul edilen bir "en iyi" algoritma yoktur. Bu çalışma, çeşitli sınıflandırma algoritmalarını birleştirmek ve tek bir sonuç elde etmek için çoğulluk kuralının kullanımını önermektedir. Çoğul oylama kuralı, en çok oy alan adayın seçimi kazandığı bir seçim sistemidir. Bu çalışmada çoğulluk kuralı iç mekânda lokalizasyon problemine uygulanmış ve "Çoğunluk Algoritması" geliştirilmiştir. Çoğunluk algoritması, beş farklı sınıflandırma algoritmasının "oylarını" alır ve çoğulluk kuralı aracılığıyla tek bir sonuç sağlar. Sonuçlar, Çoğunluk algoritmasının ortalama doğruluk oranının, oylarını kullandığı bireysel sınıflandırma algoritmalarından daha yüksek olduğunu göstermektedir. Ayrıca çalışmada, bu problem için bir sınıflandırma algoritmasının diğerinden daha iyi olduğunu beyan etmek için tek bir doğruluk oranının kullanılmasının yeterli olmadığı gösterilmiştir. Sınıflandırma algoritmaları eğitim ve test verilerini rastgele ayırmakta ve farklı veri ayrımları farklı doğruluk oranlarına sebep olmaktadır. Bu çalışmada, sınıflandırma algoritmaları karşılaştırılırken güven aralıkları kullanılmasının daha doğru bir bilgi sağladığı gösterilmektedir.**

Anahtar Kelimeler*— iç mekan lokalizasyonu, Wi-Fi sinyal gücü, sınıflandırma algoritmaları, çoğulluk kuralı

***Abstract*— User localization in an indoor environment has a wide application area including production and service systems such as factories, smart homes, hospitals, nursing homes, etc. User localization based on Wi-Fi signals has been widely studied using various classification algorithms. In this type of problem, several Wi-Fi routers placed in an indoor environment provide signals with different strengths depending on the location/room of the user. Most classification algorithms successfully make the localization with a high accuracy rate. However, in the current literature, there is no widely accepted "best" algorithm for solving this problem. This study proposes the use of the plurality rule to combine several classification algorithms and obtain a single result. Plurality voting rule is an electoral system where the candidate that polls the most vote wins the election. We apply the plurality rule to the indoor localization problem and generate the Majority algorithm. The Majority algorithm takes the "votes" of five different classification algorithms and provides a single result through plurality rule. Results show that the mean accuracy rate of the Majority algorithm is higher than the classification algorithms it combines. In addition, we show that proving a single accuracy rate is not sufficient for declaring that an algorithm is better than the other. Classification algorithms select the training and test data randomly and different divisions result in different accuracy rates. In this study, we show that comparing the classification algorithms through confidence intervals provides more accurate information.**

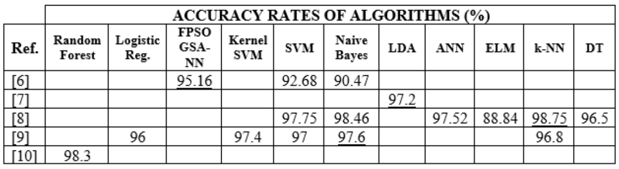
Keywords*— indoor localization, Wi-Fi signal strength, classification algorithms, plurality rule

I. INTRODUCTION

As a result of today's technological developments, use of wireless devices has become popular and indoor localization became more important [1]. Indoor localization also has a wide application area in service and production systems, including healthcare systems and factories. Healthcare systems track their assets and personnel, while production systems track their mobile tools/assets and products within a factory [2]. User localization process is more difficult in large and complex places such as factories, markets, restaurants, airports [3, 4].

There are several classification algorithms used for indoor localization, including SVM, k-NN, decision trees, random forest, artificial neural networks, and naïve bayes. There are several studies in the literature which compare these algorithms on different data sets. In this study, we use the Wireless Indoor Localization dataset of UCI Machine Learning Repository [5]. There are several studies in the literature that use the same dataset [6, 7, 8, 9, 10]. The study [6] applies artificial neural network, to improve accuracy of fuzzy hybrid of Particle Swarm Optimization & Gravitational Search Algorithm (FPSOGSA). The results are compared with PSO-NN, GSA-NN and PSOGSA-NN algorithms' results. Study shows that FPSOGSA-NN has the highest accuracy which is 95.16%. In [7] linear discriminant classifier is used for location determination. In this study z-score normalization is applied on data for faster data processing. The resulting accuracy rate is 97.2%. In [8] artificial neural network, extreme learning machine, k-NN, support vector machine, naïve bayes classifier and decision tree classification algorithms are used. Two different normalization methods are applied on data: standard score and feature scaling. The individual performance of 6 algorithms are compared with each other and k-NN algorithm has the highest accuracy value with 98.75%. In [9] logistic regression, k-NN, SVM, Kernel SVM and naïve bayes is used. Similar to [8], 2 different normalization methods are used. Before applying algorithms, linear discriminant analysis is applied on dataset for dimensionality reduction. Comparison results showed that naïve bayes has the highest accuracy with 97.6% between 5 applied algorithms. In [10] Random forest is used, and the results are compared with the results of k-NN, SVM and neural networks algorithms and it is indicated that random forest outperformed them. As a result, the accuracy of random forest model is calculated as 98.3% in the study. Table 1 summaries the results of these studies. The columns represent the accuracy rates provided by the studies and best accuracy values of applied algorithms is highlighted.

In this study, instead of just comparing classification algorithms, we combine the strengths of five different algorithms and propose the Majority algorithm. The Majority algorithm is based on the plurality voting rule which is an electoral system which elects the candidate that collected the most vote. Similarly, the Majority algorithm collects the "votes" of different classification algorithms and provides a single result based on the plurality rule. This study compares the results of the Majority algorithm with the individual algorithms. In addition, we show that the same algorithm can result in different accuracy rates when applied to different training-test partitions of the same dataset. Results show that providing a single test or mean accuracy rate can be misleading in comparing the performance of the algorithms. We show that using confidence intervals can provide more information about the actual performance of algorithms.



	ACCURACY RATES OF ALGORITHMS (%)										
Ref.	Random Forest	Logistic Reg.	FPSO GSA-NN	Kernel SVM	SVM	Naive Bayes	LDA	ANN	ELM	k-NN	DT
[6]			95.16		92.68	90.47					
[7]							97.2				
[8]					97.75	98.46		97.52	88.84	98.75	96.5
[9]		96		97.4	97	97.6				96.8	
[10]	98.3										

Table I. Accuracy rates of same dataset used in different studies

II. Materials and Method

This section introduces the methods used in the Majority algorithm and he test procedure. Section A introduces the 5 algorithms that the Majority Algorithm combines. Section B introduces the Majority algorithm, and Section C describes the test procedure. All algorithms in this section are coded in Python.

A. Methods

1) Logistic Classifier: Logistic regression is based on statistics and is an extension of linear regression models. Logistic regression is used for classification problems with two outcomes and models the probabilities of each outcome [11]. Multinomial logistic regression, also known as soft max regression, is a supervised learning algorithm that can be used in a variety of problems, including the subject being studied, due to the hypothesis function it uses [13]. Given a set of independent variables, Multinomial logistic model allows for more than two categories of the outcome/dependent variable [14].

In this study, the logistic classifier in the turicreate library [12] was used with parameters: l1 penalty is equal to 0, l2 penalty is equal to 0.01, solver is auto selected, l2-norm rescaling is performed, convergence threshold is equal to 0.01, step size is equal to 1.0, lbfgs memory level is set to 11, max iterations is set to 10, class weights are equal.

2) Decision Tree: One of the most common ML algorithms for regression and classification problems is decision tree algorithm [15]. This algorithm can model nonlinear interactions between properties and the target, unlike linear models such as logistic regression or SVM. Decision tree algorithm has ability to deal with continuous data as well as categorical data.

In this algorithm, the decision tree is a structure in which nodes point attributes, links point decision rules, and leaves point output classes. The aim is to introduce a tree-like structure for input attributes and to create a unique output on each leaf. [15,16]. It is a special case of gradient boosted trees algorithm whose number of trees set to 1. In this study, the decision tree in the turicreate library [17] was used with parameters: class weights are equal, maximum depth of tree is set to 6, minimum loss reduction is set to 0, minimum weight of each leaf node is set to 0.1.

3) Boosted Tree: In many implementations, including multi-class classification, flocculation process modeling, rank and click prediction, the gradient-boosted decision tree is a strong machine learning algorithm frequently used. [18]. Similar to other ensemble models, gradient boosted tree algorithm also uses decision trees the same way; each successive estimator tries to reduce the error of previous step which is gradually generated [19].

It starts with a basic decision tree model being trained on a dataset. Inaccurately predicted observations are set aside, and then a specific model is trained that is not guaranteed to correctly estimates other samples as this process continues, models are weighted according to the level of difficulty of the samples they correctly predict. All models are combined in a single model or boosted trees if training performance is near as iterations advance. [20]. Unlike deep learning GBDT is more flexible, efficient and a user-friendly open source toolkit [21]. Unlike linear models such as logistic regression or SVM, gradient boosted trees can model nonlinear interactions between inputs and the output. This model is suitable for both categorical and numerical properties.

In this study, the boosted trees in the turicreate library [22] was used as a gradient boosted trees algorithm with parameters: maximum number of iterations is equal to 60, class weights are equal, maximum depth of a tree is set to 6, step size is equal to 0.3, minimum loss reduction is equal to 0, minimum weight of each leaf node is set to 0.1, the ratios of subsample row and column both are equal to 1.0 which means all base trees uses all training set and all features of it.

4) k-NN: K-Nearest Neighbour (k-NN) is a simple and widely used algorithm which is non-parametric and supervised learning method that can be used for classification. The algorithm tries to find k closest training examples based on their Euclidean distances between each data points. The neighbors are selected from the data set which the classes are known. Therefore, the only data that are needed to train the model is the training data which is taken from the data. No further steps are needed. In the k-NN algorithm, the class of test data is determined by looking at the class of closest train data to the test data. The nearest class is selected for test data [8].

The number of class is selected as 4 considering the fact that the data have 4 rooms. The model is built by KNeighborsClassifier which is in the scikit-learn library.

5) Support Vector Machine:

Finding a hyperplane or hyperplanes that are decision boundaries that help classify the data points in multi-dimensional space is the main point of Support Vector Machine (SVM). The algorithm helps us to solve how hyperplanes should be drawn. To use this algorithm, the data must consist more than one class [23]. If the data consist of two classes, our data can be separated by a line, which is the farthest line to the data points. If the data consist of n classes, the hyperplane will be a (n-1) dimension plane.

B. Majority Algorithm

Plurality voting is an electoral system. In this system each voter can vote for only one candidate, and the candidate who gets the most vote among all candidates is elected. The Majority algorithm is a simple algorithm that only combines the results of other classification algorithms. In Majority Algorithm, each algorithm Section 2.1 describes (Logistic Classifier, Decision Tree, Boosted Tree, k-Nearest Neighbor, Support Vector Machine) has one vote. Given the same problem instance data, each algorithm makes an estimation – gives a “vote” for the room the user is localized. Majority algorithm counts the votes and selects the room that gets the most vote. If two or more rooms get the same “most” number of votes, then the Majority algorithm picks one of these rooms randomly. The Majority algorithm can be applied to any other selection of classification algorithms. Section 2.3 describes the test procedure and Section 3 compares the performance of the algorithms.

C. Test Procedure

This study uses the Wireless Indoor Localization dataset of UCI Machine Learning Repository [5]. The dataset provides Wi-Fi signal strengths measured in 4 different rooms, coming from 7 different routers. There are a total of 2000 instances in the dataset. The same data set is used in various studies in the literature, including [6].

When the training and test partitions are randomly selected from a dataset, different random selections can result in different accuracy rates for the same algorithm. Comparing algorithms based on a single test set does not provide reliable results. To make a reliable comparison, we followed the test procedure below and computed upper and lower bounds for 95% confidence intervals of mean accuracy rate. In this procedure we call the Logistic Classifier (LC), Decision Tree (DT), Boosted Tree (BT), k-Nearest Neighbor (kNN), Support Vector Machine (SVM) as “main” algorithms.

Test procedure:

Step 0. Set the iteration number. iterNumber = 1

Step 1. Divide the data set into two partitions randomly: 70% training, 30% test

Step 2. Use the training partition and train the main algorithms separately with the same training data set.

Step 3. Use the 30% test partition and test each algorithm. For each instance on the test set, collect the class estimations from the 5 main algorithms. Apply the Majority algorithm on these and obtain the class estimation of the Majority algorithm.

Step 4. Compute the accuracy rate of these six algorithms (5 main + Majority). Record the accuracy rate of each algorithm separately.

Step 5. If iterNumber=100 go to Step 6. Else, iterNumber= iterNumber+1 and go to Step 1.

Step 6. Compute 95% confidence intervals for the mean accuracy rates of all algorithms. Find the minimum and maximum accuracy rates of these 100 iterations for each algorithm.

III. RESULTS AND DISCUSSION

This section presents the accuracy rates of the main algorithms Logistic Classifier (LC), Decision Tree (DT),

Boosted Tree (BT), k-Nearest Neighbor (kNN), Support Vector Machine (SVM) and the proposed Majority algorithm. Table 2 shows the mean accuracy values, minimum and maximum accuracy values of 100 iterations for each algorithm, standard deviations and 95% confidence interval upper and lower bounds of accuracy values. Accuracy rates is calculated by using normalized data values. The normalization method is feature scaling. Figure 1 shows the mean accuracies and 95% confidence intervals of mean accuracies. It can be seen from Table 2 and Figure 1 that according to mean accuracies the best value is 98.36% by Majority Algorithm. On the other hand, mean accuracy of k-NN is very close to the Majority Algorithm. Although, Majority Algorithm has better mean accuracy, maximum accuracy value of k-NN is higher than Majority Algorithm. However, the minimum value of majority algorithm is better than that of k-NN.

As these results show, comparing the accuracy of the algorithms obtained from a single iteration can often be misleading. Because of the closeness between k-NN and Majority Algorithm results, to be able to understand which one is better than other, paired t test is applied on k-NN and Majority algorithms. The 95% confidence interval of the difference between the mean accuracy rates of k-NN and the Majority algorithm is calculated as [-0.045, 0.045]. As this interval includes 0, we cannot say that one algorithm is superior than the other. When the minimum and maximum accuracy values are inspected, we can see that the maximum accuracy of 99.5% is achieved by k-NN in one of the 100 iterations. This accuracy rate is higher than the accuracy rates obtained by the other studies on the same dataset in the literature (highest accuracy rate of 98.75% reported by [8]). This result once again shows that random separation of the training and test datasets effects the reported performance of the algorithms considerably and performance of a single run is not a reliable indicator of the performance of an algorithm.

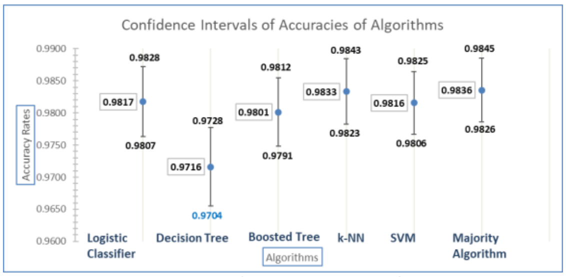


Figure 1. 95% Confidence intervals of algorithms

Statistics	LC	DT	BT	k-NN	SVM	Majority
Average	0.982	0.972	0.980	0.983	0.982	0.984
Std. Deviation	0.005	0.006	0.005	0.005	0.004	0.005
Minimum Accuracy Value	0.967	0.953	0.965	0.967	0.967	0.970
Maximum Accuracy Value	0.993	0.987	0.990	0.995	0.993	0.992
Lower bound	**0.981**	**0.970**	**0.979**	**0.982**	**0.981**	**0.983**
Upper bound	**0.983**	**0.973**	**0.981**	**0.984**	**0.982**	**0.984**

Table II. Statistics of algorithms applied

IV. CONCLUSION

Wireless indoor localization is widely used in indoor systems such as smart buildings, factories, hospitals and nursing homes. Wireless indoor localization is especially important in the Industry 4.0 area. Industry 4.0 focuses on technologies such as machine to machine communication and internet of things to enable increased automation in factories. Wireless indoor localization systems automatically locate mobile tools/assets and products in the factories, increasing the speed, transparency and efficiency of production systems.

There exist several classification algorithms used for wireless indoor localization and all have different accuracy values on different data sets. This study proposes the Majority algorithm which uses the plurality rule for combining the information gathered from different classification algorithms. The Majority algorithm runs several different classification algorithms on the same problem, collects their “votes” on the class estimates and selects the class that was most voted for.

Results show that the Majority algorithm has the best mean accuracy rate over 100 iterations. However, as the dataset’s test and training partitions are selected randomly, the results also contain randomness and therefore should be expressed with confidence intervals. The Majority algorithm is promising with its high mean and minimum (of 100 iterations) accuracy rates. This algorithm should be tested with several different datasets and main algorithms to prove its value. Future studies should concentrate on testing the value of Majority algorithm with different problem areas and datasets where the classification algorithms alone provide lower accuracy rates.

References

[1] F. Zafari, A. Gkelias, and K. Leung, "A Survey of Indoor Localization Systems and Technologies," ar Xiv preprint arXiv:1709.01015, 2017.

[2] Fescioglu-Unver, N., Choi, S. H., Sheen, D., & Kumara, S. (2015). RFID in production and service systems: Technology, applications and issues. Information Systems Frontiers, 17(6), 1369-1380.

[3] S. Xia, Y. Liu, G. Yuan, M. Zhu, and Z. Wang, "Indoor fingerprint positioning based on WiFi: an overview," ISPRS International Journal of Geo-Information, vol. 6, no. 5, p. 135, 2017.

[4] G. Sithole and S. Zlatanova, "Position, Location, Place and Area: An Indoor Perspective," ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 3, p. 89, 2016.

[5] UCI Machine Learning Repository. Available: https://archive.ics.uci.edu/ml/index.php Last accessed: 23.09.2020

[6] J. G. Rohra, B. Perumal, S. J. Narayanan, P. Thakur, and R. B. Bhatt, "User localization in an indoor environment using fuzzy hybrid of particle swarm optimization & gravitational search algorithm with neural networks," in Proceedings of Sixth International Conference on Soft Computing for Problem Solving, 2017, pp. 286-295: Springer.

[7] O. Altay and M. Ulas, "Location determination by processing signal strength of Wi-Fi routers in the indoor environment with linear discriminant classifier," 2018 6th International Symposium on Digital Forensic and Security (ISDFS), Antalya, 2018, pp. 1-4, doi: 10.1109/ISDFS.2018.8355353.

[8] K. Sabanci, E. Yigit, D. Ustun, A. Toktas and M. F. Aslan, "WiFi Based Indoor Localization: Application and Comparison of Machine Learning Algorithms," 2018 XXIIIrd International Seminar/Workshop on Direct and Inverse Problems of Electromagnetic and Acoustic Wave Theory (DIPED), Tbilisi, 2018, pp. 246-251, doi: 10.1109/DIPED.2018.8543125.

[9] M. Kumar, M. Rawat and P. G. Shambharkar, "Localization of User in an Indoor Environment Using Machine Learning Classification Models," 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2020, pp. 714-719, doi: 10.1109/ICICCS48265.2020.9121134.

[10] R. Gomes, M. Ahsan and A. Denton, "Random Forest Classifier in SDN Framework for User-Based Indoor Localization," 2018 IEEE International Conference on Electro/Information Technology (EIT), Rochester, MI, 2018, pp. 0537-0542, doi: 10.1109/EIT.2018.8500111.

[11] Molnar, C. (2020). Interpretable machine learning—A guide for making black box models explainable. https://christophm.github.io/interpretable-ml-book/.

[12] Turicreate.logistic_classifier.LogisticClassifier (n.d.). Retrieved August 06, 2020, from https://apple.github.io/turicreate/docs/api/generated/turicreate.logistic_classifier.LogisticClassifier.html

[13] P. S. Br Ginting, B. Irawan and C. Setianingsih, "Hate Speech Detection on Twitter Using Multinomial Logistic Regression Classification Method," 2019 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS), BALI, Indonesia, 2019, pp. 105-111, doi: 10.1109/IoTaIS47347.2019.8980379.

[14] Hosmer Jr, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). Applied logistic regression (Vol. 398). John Wiley & Sons.

[15] Harikumar Rajaguru, Sannasi Chakravarthy S R (2019) Analysis of Decision Tree and K-Nearest Neighbor Algorithm in the Classification of Breast Cancer, Asian Pac J Cancer Prev. 2019; 20(12): 3777–3781, doi: 10.31557/APJCP.2019.20.12.3777

[16] Byunghoon Kim, Young-Seon Jeong, Seung Hoon Tong & Myong K. Jeong (2020) A generalised uncertain decision tree for defect classification of multiple wafer maps, International Journal of Production Research, 58:9, 2805-2821, DOI: 10.1080/00207543.2019.1637035

[17] Turicreate.decision_tree_classifier.DecisionTreeClassifier (n.d.). Retrieved August 06, 2020, from https://apple.github.io/turicreate/docs/api/generated/turicreate.decision_tree_classifier.DecisionTreeClassifier.html

[18] Zhang, Zhendong, ve Cheolkon Jung. “GBDT-MO: Gradient Boosted Decision Trees for Multiple Outputs”. arXiv:1909.04373 [cs], December 2019. arXiv.org, http://arxiv.org/abs/1909.04373.

[19] G. Donkal and G. K. Verma, "Securing Big Data Ecosystem with NSGA-II and Gradient Boosted Trees Based NIDS Using Spark," 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2018, pp. 146-151, doi: 10.1109/ICCONS.2018.8663120.

[20] A. Lasisi, M. O. Sadiq, I. Balogun, A. Tunde-Lawal and N. Attoh-Okine, "A Boosted Tree Machine Learning Alternative to Predictive Evaluation of Nondestructive Concrete Compressive Strength," 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), Boca Raton, FL, USA, 2019, pp. 321-324, doi: 10.1109/ICMLA.2019.00060.

[21] Z. Wen, B. He, R. Kotagiri, S. Lu and J. Shi, "Efficient Gradient Boosted Decision Tree Training on GPUs," 2018 IEEE International Parallel and Distributed Processing Symposium (IPDPS), Vancouver, BC, 2018, pp. 234-243, doi: 10.1109/IPDPS.2018.00033.

[22] Turicreate.boosted_trees_classifier. BoostedTreesClassifier (n.d.). Retrieved August 06, 2020, from https://apple.github.io/turicreate/docs/api/generated/turicreate.boosted_trees_classifier.BoostedTreesClassifier.html

[23] Support Vector Machine (SVM) Algorithm - Javatpoint. (n.d.). Retrieved August 06, 2020, from https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm