

PERFORMANCE ANALYSIS OF CLASSIFICATION METHODS FOR INDOOR LOCALIZATION IN VLC NETWORKS

D. Sánchez-Rodríguez ^{a,*}, I. Alonso-González ^a, J. Sánchez-Medina ^b, C. Ley-Bosch ^a, L. Díaz-Vilarino ^{c,d}

^a Institute for Technological Development and Innovation in Communications, University of Las Palmas de Gran Canaria, Campus de Tafira, CP 35017, Las Palmas de Gran Canaria, Spain - (david.sanchez, itziar.alonso, carlos.ley)@ulpgc.es

^b Institute for Cybernetics, University of Las Palmas de Gran Canaria, Campus de Tafira, CP 35017, Las Palmas de Gran Canaria, Spain – javier.sanchez@ulpgc.es

^c Applied Geotechnologies Group, Dept. Natural Resources and Environmental Engineering, University of Vigo, Campus Lagoas-Marcosende, CP 36310 Vigo, Spain - lucia@uvigo.es

^d TU Delft - Faculty of Architecture, OTB, section GIS Technology, Delft, Netherlands – L.Diaz-Vilarino@tudelft.nl

Commission IV, WG IV/5

KEY WORDS: Indoor Localization, Visible Light Communication, Machine Learning Classifiers, Fingerprinting Techniques

ABSTRACT:

Indoor localization has gained considerable attention over the past decade because of the emergence of numerous location-aware services. Research works have been proposed on solving this problem by using wireless networks. Nevertheless, there is still much room for improvement in the quality of the proposed classification models. In the last years, the emergence of Visible Light Communication (VLC) brings a brand new approach to high quality indoor positioning. Among its advantages, this new technology is immune to electromagnetic interference and has the advantage of having a smaller variance of received signal power compared to RF based technologies. In this paper, a performance analysis of seventeen machine learning classifiers for indoor localization in VLC networks is carried out. The analysis is accomplished in terms of accuracy, average distance error, computational cost, training size, precision and recall measurements. Results show that most of classifiers harvest an accuracy above 90%. The best tested classifier yielded a 99.0% accuracy, with an average error distance of 0.3 centimetres.

1. INTRODUCTION

Indoor localization has been a term of growing interest over the past decade as lightweight mobile devices have become the standard in the real world. Many user applications for these devices need some notion of the current position, and hence, the development of localization techniques is one of the keys to the success of pervasive computing. Thus, location-aware services have made it possible to use applications capable of sensing their location and modifying their setting and functions accordingly (Want, 2001).

Many indoor localization approaches based on globally deployed radiofrequency communication systems, such as WLAN, Bluetooth and UWB, have been proposed, mainly because of their low cost and mature standardization state. In these systems, the fingerprinting technique is one of the most commonly used for indoor localization (Honkavirta, 2009). This kind of technique estimates positioning by matching online measured data with pre-measured location-related data, such as received signal strength (RSS). Hence, just RSS information is needed and extra sensors are unnecessary. Localization based on fingerprinting is usually carried out in two phases. In the first phase, normally termed offline phase, a database of the RSS samples is built from different base stations at each reference location for the target environment. Using those samples as a training set, a positioning model is learnt using a particular machine learning technique. In this phase, it can be found a great

diversity on the applied methodologies. During the second phase, namely the online phase, the location is determined by means of new RSS measurements collected in a specific position and using the learnt model in the previous phase.

A major drawback of fingerprinting techniques is that the key parameter (RSS) for predicting the position of a device is not stable with time due to dense indoor multipath effects, such as reflection, diffraction and scattering. Multipath fading causes the received signal to fluctuate around a mean value at a particular location (Kaemarungsi, 2012). Therefore, they usually deliver an accuracy of up to two meters, since they are hindered by multipath propagation.

On the other hand, VLC is experiencing a growing interest because of improvements in solid state lighting and a high demand for wireless communications. Although line of sight is necessary for efficient communications in VLC networks, this kind of network infrastructure can offer a higher positioning accuracy mainly because of two reasons: it is not affected by electromagnetic interferences and the received optical power is more stable than radio signals, so it can be accurately determined (Armstrong, 2013). Therefore, fingerprinting techniques are expected to yield higher accuracy in VLC networks. In this paper, a performance analysis of different machine learning classifiers using RSS samples is carried out. RSS values are obtained using a VLC simulator that implements the IEEE 802.15.7 standard. To be precise, six classifiers are studied,

* Corresponding author

namely K-Nearest Neighbour, Random Forest, C4.5, REPTree, KStar and LMT. Furthermore, Boosting and Bagging techniques are also analysed using the previous classifiers as “weak learners”. Hence, seventeen classifiers are analysed in this paper. The analysis is carried out in terms of accuracy, average distance error, computational cost, training size, precision and recall measurements.

Within the last few years, many studies on VLC based positioning have been published. Nevertheless, to the best of our knowledge, to this date there are no published indoor positioning research papers where a performance analysis of different classifiers is carried out.

The rest of the paper is organized as follows. In Section 2, we describe our simulator that implements the IEEE 802.15.7 standard for VLC networks. Next, in Section 3, machine learning classifiers used in this paper are briefly described. In Section 4, the test environment is defined. In Section 5, the evaluation of specific parameters on the performance of classifiers is shown. Finally, the conclusions are summed up and future works are presented.

2. SIMULATOR DESCRIPTION

In our research group, we have developed a simulator for IEEE 802.15.7 networks. It was developed using OMNET++ (Omnet, 2009) simulation framework from the model developed by (Chen, 2008) designed for sensor networks based on the IEEE 802.15.4 standard, because of the similarities existing between IEEE 802.15.7 and IEEE 802.15.4 architectures.

OMNeT++ provides built-in support tools not only for simulation, but also for the analysis and visualization of results. Several data types can be used to analyse simulation results, such as throughput, delay, packet loss and RSS. In this paper, the simulator is used to obtain RSS samples in a receiver grid acquired from the signal coming from different emitters (also called coordinators). These RSS samples are used as features for training the classification methods.

The developed simulation model has been designed with the following premises:

- IEEE 802.15.7 star topology has been chosen because of its importance and wide range of applications.
- For the MAC layer, we opted to use the superframe structure; since it allows the use of both contention (CAP) and no contention (CFP) access methods. In addition, the use of the superframe enables devices to enter the energy save state during the idle period.
- A VPAN identifier is assigned to each emitter to identify each coordinator (LED lamp).

In the next subsections, the most important features in our simulator is described, for a better comprehension of the presented results.

2.1 Optical channel model

The transmission medium is modelled as free space without obstacles. The directed line of sight (LOS) link configuration was chosen to model the optical signal propagation, requiring a

LOS between each device and the coordinator. Only the direct component of the received signal was considered to calculate the received power, neglecting the possible influence of reflections.

Frequency response of optical channel is relatively flat near Direct Current (DC), so the most important quantity for characterizing this channel is the DC gain $H(0)$ (Kahn, 1997), which relates the transmitted and received optical average power, see Equation 1:

$$P_r = H(0) \times P_t \quad (1)$$

In VLC, the received power can be expressed as the sum of LOS and non-LOS components. In directed LOS links, the DC gain can be computed accurately by considering only the direct LOS propagation path. According to the results presented in (Komine, 2004), at least 90% of total received optical power is direct light in VLC when using a receiver field of view (FOV) of 60 degrees. Figure 1 shows an example of a directed LOS link.

An optical source can be modelled by its position vector, a unit-length orientation vector \vec{o}_t , transmission power P_t and a radiation intensity pattern $I(\theta, m)$ emitted in direction θ . Where m is the mode number of the radiation lobe, which specifies the directionality of the source, and is related to the transmitter half power angle $\theta_{1/2}$. Similarly, a receiver is defined by its position, orientation \vec{o}_r , photo detector area A , and FOV (ψ_c). The angle formed between the optical incident signal and the orientation vector \vec{o}_r is called the incident angle ψ . The maximum incident angle defines the receiver FOV. Considering LOS propagation path, the DC gain can be calculated according to (Kahn, 1997) as Equation 2:

$$H(0) = \frac{(m+1) \times \cos^m(\theta) \times A \times T_s(\psi) \times G(\psi) \times \cos(\psi)}{2 \times \pi \times d^2}, \quad 0 \leq \psi \leq \psi_c$$

$$H(0) = 0, \quad \psi > \psi_c$$

where $T_s(\psi)$ = optical filter signal transmission coefficient
 $G(\psi)$ = optical concentrator gain
 d = distance between transmitter and receiver

The adopted optical channel model facilitates reaching high

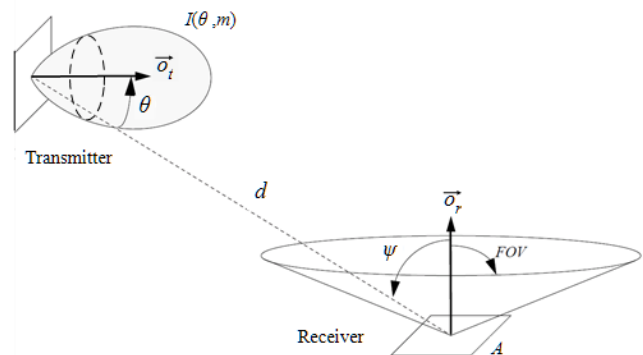


Figure 1. Directed LOS link configuration.

transmission speeds, since the effects of multipath distortion on the optical signal are not considered. Considering only the direct component of the signal has the additional benefit of improving the efficiency of the implemented simulation model. The computational load required to run simulations of scenarios with multiple nodes including the functionality of different layers of the architecture is reduced significantly.

To ensure the validity of our implemented model, we have configured all optical receivers using a 60 degrees FOV value (ψ_c).

2.2 PHY layer simulation parameters

Table 1 shows the main configuration parameters of PHY layer used in all simulation scenarios. We selected the PHY II operating mode, intended for both indoor and outdoor environments, using MCS-ID number 16, since support for the minimum clock and data rates for a given PHY is mandatory.

Because of the optical channel model used, transmitters' directivity is characterized by its half power angle, $\theta_{1/2}$ while receivers' directivity is defined by its FOV. According to (Chvojka, 2015), both parameters are assigned a value of 60 degrees, to ensure validity of the implemented channel model, since the calculation of received optical power takes into account only the direct component of the signal.

In order to simplify the calculation process of the model, the values used for the concentrator gain ($G(\psi)$) and the transmission coefficient of the optical filter ($T_s(\psi)$) are set up as constant values, so they do not depend on the angle of incidence ψ .

The rest of the selected values employed to characterize VLC transmitters and receivers are commonly used values in literature, similar to those used in (Chvojka, 2015)(Tronghop, 2012).

| Parameter | Value |
|---|---------------------|
| Transmission rate | 1.25 Mbps |
| Optical clock rate | 3.75 MHz |
| Coordinator optical transmission power (P_t) | 15 W |
| Half Power Angle $\theta_{1/2}$ | 60° |
| Field of Vision (ψ_c) | 60° |
| Photo detector area (A) | 100 mm ² |
| Photo detector responsivity (R) | 0.54 A/W |
| Optical concentrator gain ($G(\psi)$) | 15 |
| Optical filter transmission coefficient ($T_s(\psi)$) | 1 |

Table 1. PHY layer parameters.

3. MACHINE LEARNING CLASSIFIERS

In this section, a brief description of used classifiers in this paper is outlined.

3.1 K-Nearest Neighbour

KNN is a machine learning algorithm that predicts the classification of new data based on the closest training samples in the feature space (Cover, 1967). The algorithm decides which class is similar by picking the K nearest data point distances to

the observation. Then, simple majority of neighbours is used to determine the class prediction. In this paper, IB1 implementation (Aha, 1991) of K-NN was used and the number of nearest neighbours was established in K=1 because this configuration provided better results.

3.2 Random Forest

RandomForest (RF) was proposed by Breiman (Breiman, 2001). Random Forest is an ensemble of decision trees such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. As the number of trees in the forest becomes large, the generalization error converges to a limit. The generalization error of a forest of tree classifiers depends on the strength of the individual trees in the forest and the correlation between them. The classification is done by a majority vote among the decisions of all trees. The randomness introduces robustness to the algorithm against noise and outliers. Random Forest is equally applicable to both classification and regression problems. In this paper, the number of trees was established in 100.

3.3 C4.5

C4.5 is an algorithm used to generate a decision tree (Quinlan,1993). C4.5 builds decision trees from a set of training data using the concept of information entropy. At each node of the tree, C4.5 chooses the attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other. The splitting criterion is the normalized information gain (difference in entropy). The attribute with the highest normalized information gain is chosen to make the decision. The C4.5 algorithm then recurs onto the smaller sublists. In this paper, a confidence factor equals to 0.25 was used because better results were reached.

3.4 REPTree

Reduced Error Pruning Tree is a fast decision tree learning and it builds a decision tree based on the information gain (Srinivasan, 2014). REP Tree builds a decision/regression tree using information gain as the splitting criterion, and prunes it using reduced error pruning. It only sorts values for numeric attributes once.

3.5 LMT

A Logistic Model Tree basically consists of a standard decision tree structure with logistic regression functions at the leaves (Landwehr, 2005), much like a model tree is a regression tree with regression functions at the leaves. It combines the logistic regression models with tree induction, and thus is an analogue of model trees for classification problems. As in ordinary decision trees, a test on one of the attributes is associated with every inner node. For a nominal attribute with k values, the node has k child nodes, and instances are sorted down one of the k branches depending on their value of that attribute. For numeric attributes, the node has two child nodes and the test consists of comparing the attribute value to a threshold: an instance is sorted down the left branch if its value for that attribute is smaller than the threshold and sorted down the right branch otherwise.

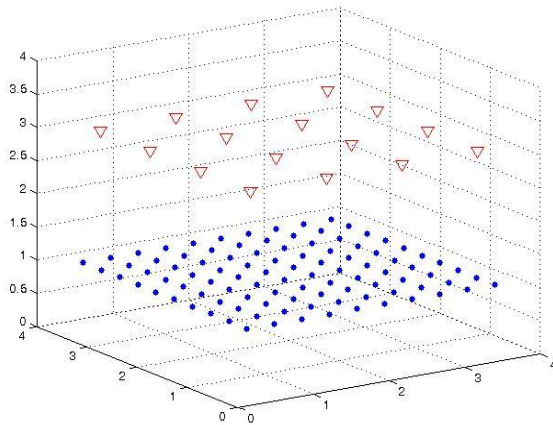


Figure 2. Scenario with 16 LED lamps and 100 receivers.

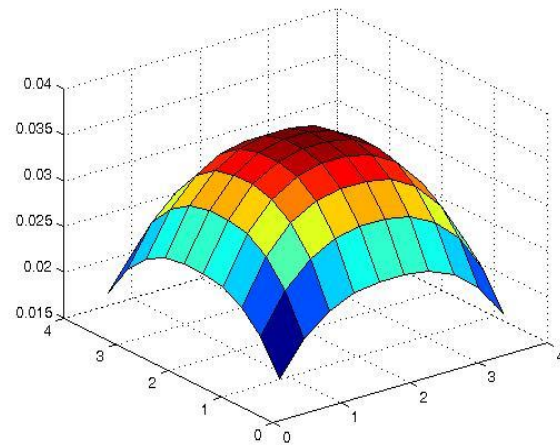


Figure 3. Distribution of the received optical power at 1 metre from the floor.

3.6 KStar

K* is an instance-based classifier, that is the class of a test instance is based upon the class of those training instances similar to it, as determined by some similarity function. It differs from other instance-based learners in that it uses an entropy-based distance function. (Cleary, 1995).

3.7 Boosting

The boosting method is a technique to improve the classification accuracy of tree based classifiers. The idea of boosting is to combine the prediction of many base or weak classifiers to form a powerful classifier. AdaBoost is the most popular boosting algorithm used for classification (Freund, 1996). It is an adaptive and iterative algorithm that combines base models of the same type, such as a C4.5 decision tree, in such a way that each new base model is influenced by the performance of those base models built in previous iterations. In this paper, we use all aforementioned classifiers as base model, except the K-NN classifier because boosting is not possible with K=1. The algorithm was iterated 10 times.

3.8 Bagging

Bagging (Bootstrap Aggregating) was proposed by Breiman (Breiman, 1996) to improve the classification by combining classifications of randomly generated training sets. By increasing the size of the training set the model predictive force cannot be improved, but just decrease the variance, narrowly tuning the prediction to expected outcome. It also helps to avoid overfitting. Although it is usually applied to decision tree methods, it can be used with any type of method. In this paper, we use all aforementioned classifiers as base model. The algorithm was iterated 10 times.

4. TEST ENVIRONMENT

In this section, we describe the test environment used to evaluate the performance of machine learning classifiers described in Section 3. The simulation environment configured in our IEEE 802.15.7 simulator models a 4 by 4 by 3 metres room. The scenario is shown in Figure 2. This environment consists of 16

coordinators or LED lamps (red triangles), configured as a 4 x 4 grid placed 1 meter apart from each other on the ceiling. On the lower part, we set up 100 receivers (blue circles) in a 10 x 10 grid configuration, with a 36 centimetres separation from each other. In order to evaluate the effects of having different distances between receivers and coordinators, the receivers plane is set up at three different heights: 75, 100 and 125 centimetres from the floor. Receivers orientation was randomly assigned for each simulation as follows: they are pointing out to the ceiling with an initial orientation vector [0,0,1] and a random offset (-0.2,0.2) is applied to each axis in each simulation. Thus, each receiver has a different orientation in each simulation.

Eleven simulations were performed on each one of three aforementioned receiver planes. One RSS measurement from each LED lamp was estimated at each receiver in every simulation. This leads to 3,300 (11 samples x 3 layers x 100 receivers) RSS measurements from each LED lamp. Hence, the dataset is finally composed of 3,300 instances, where each instance stores the RSS samples from each LED lamp estimated in a receiver. Figure 3 shows the received optical power (lux) at 1 metre from the floor with sixteen coordinators. It shows that there is enough lighting to receive the beacon frame in every reference location. The simulation parameters were specified in Table 1.

5. EXPERIMENTAL RESULTS AND DISCUSSION

In order to evaluate the performance of the machine learning classifiers on VLC networks, the WEKA machine learning tool (Hall, 2009) was used. Weka is an open source collection of machine learning algorithms for data mining tasks, more specifically data pre-processing, clustering, classification, regression, visualization and feature selection.

Experiments were focused onto comparing accuracy, error distance, computation time, training size, precision and recall measurements by different classifiers. The error is the expected distance from the misclassified instance (estimated receiver) and the real location (real receiver). The error is calculated by the Euclidean distance between these points, and the arithmetic mean was computed from the results of the experiments. Being a

classification problem, an error simply means that a receiver was estimated to be in a wrong positioning cell, in the receiver's grid. All experiments were carried out on an Intel Core i7 3.4 GHz/32 GB RAM non-dedicated Windows machine.

5.1 Accuracy of classifiers and computational cost

In this section, the performance of classifiers is analysed. For the validity of experimental results, the experiments were carried out using 10-fold cross-validation.

Table 2 shows the accuracy, error distance and the time to build the analysed classifiers, that is, training time. As can be seen, all classifiers have an accuracy above 90%, except REPTree algorithm. K-NN classifier obtains the best result, yielding a 99.0% accuracy, with an average error distance of 0.3 centimetres, 6 times less than the next best classifier, KStar. Furthermore, K-NN algorithm is the fastest to build the classifier.

On the other hand, Boosting and Bagging techniques outperform the performance of base classifiers C4.5, REPTree and LMT, but at the expense of much higher computation effort. Furthermore, it is noticed that Boosting techniques are slightly more accurate than Bagging techniques.

Figure 4 shows the cumulative distribution function (CDF) for the best analysed classifiers, that is, K-NN, Random Forest, AdaBoost C45, AdaBoost RepTree, KStar and AdaBoost LMT. As can be seen, most of the test instances are correctly classified, and most of the misclassified instances are about 36 centimetres, that is, these instances are the nearest neighbours (receivers) of exact locations in the same height. On the other hand, the maximum error of K-NN and KStar classifiers is about 50 centimetres.

| Classifier | Accuracy (%) | Error Distance (cm) | Training Time (s) |
|------------------|--------------|---------------------|-------------------|
| K-NN | 99.00 | 0.3 | 0.02 |
| Random Forest | 95.33 | 2.1 | 9.08 |
| C.45 | 91.06 | 6.7 | 0.30 |
| REPTree | 85.84 | 12.2 | 0.32 |
| KStar | 97.21 | 1.8 | 0.02 |
| LMT | 96.30 | 4.4 | 167.10 |
| AdaBoost RF | 95.12 | 2.1 | 8.77 |
| AdaBoost C4.5 | 95.18 | 2.2 | 7.43 |
| AdaBoost REPTree | 95.21 | 3.5 | 6.99 |
| AdaBoost KStar | 96.84 | 2.3 | 2313.39 |
| AdaBoost LMT | 96.36 | 3.3 | 1527.98 |
| Bagging K-NN | 98.61 | 0.5 | 0.03 |
| Bagging RF | 94.66 | 2.3 | 64.57 |
| Bagging C4.5 | 94.54 | 3 | 1.93 |
| Bagging REPTree | 93.24 | 3.7 | 2.28 |
| Bagging KStar | 96.57 | 2.3 | 0.02 |
| Bagging LMT | 95.97 | 3.8 | 1677.94 |

Table 2. Performance of classifiers.

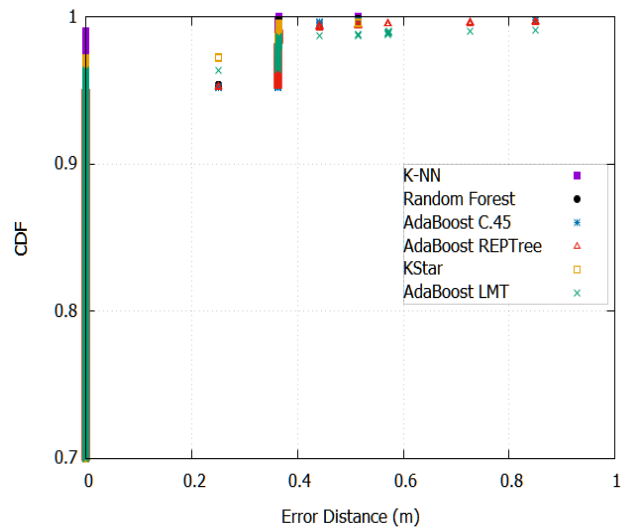


Figure 4. CDF of performance for the best analysed classifiers.

5.2 Analysis of Training Dataset size

The training dataset size is an important parameter for the performance and the building time of each model based on decision trees. A large-sized training dataset can provide better accuracy to predict the correct location, but too much data can increase the elapsed time to build the model considerably. The aim is to reduce as much as possible the training phase achieving a minimal impact on the performance. In order to test the robustness of the method, different training dataset sizes were used, from 20% to 80% of the whole dataset. For the validity of experimental results, the experiments were performed 100 times, each time selecting the training and testing data after randomizing the instances order, picking the same proportion of samples at each class (stratified split).

Table 3 shows the experimental results for different training sizes for the best classifiers in the previous section, except AdaBoost LMT, which is replaced by LMT because this latter has similar accuracy and a considerably lower computational cost. As can be seen, the performance of analysed algorithms varies on training set size, and the accuracy improves with larger training datasets. Furthermore, an error distance below 10 centimetres is yielded using a 60% training size for all classifiers. Although a similar value of accuracy is obtained with only a 20% training dataset size when using K-NN algorithm. This error distance may be enough for some location based applications. Regarding the training time, it should be noted that the Random Forest, LMT and Boosting based classifiers take more computation time than K-NN and KStar algorithms, at least 500-times worse using a 60% training size.

Precision and recall measures have been also evaluated with training size. Figures 5 and 6 show the precision and recall measures, respectively. As can be seen, results for both measures follow a similar trend compared with accuracy, that is, they increase when the training size does, yielding the K-NN classifier the best results with both values close to 0.99 using an 80% of training size.

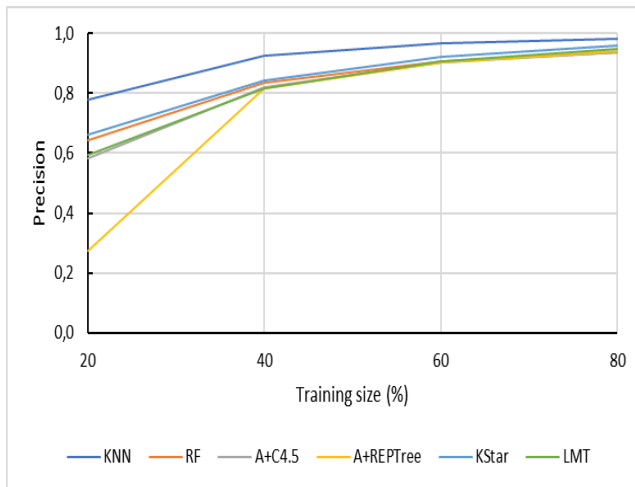


Figure 5. Precision

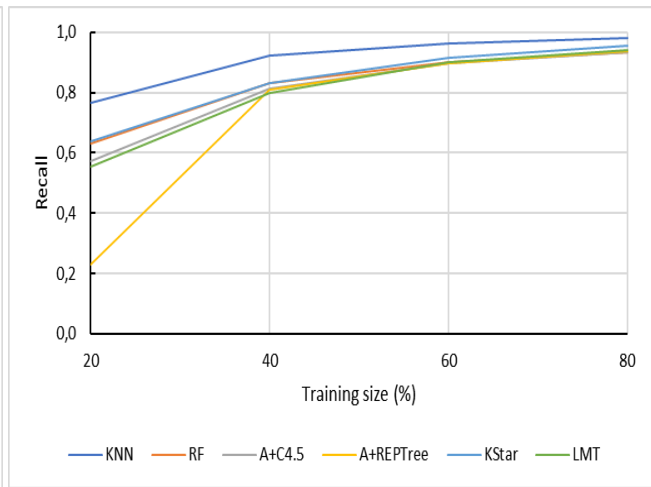


Figure 6. Recall

6. CONCLUSIONS

In this paper, we have analysed the performance of different machine learning classifiers for indoor localization in VLC networks. Accuracy, error distance, computational cost, training size, precision and recall measurements were evaluated.

Regarding accuracy, most of the analysed classifiers yielded excellent results, above 90% accuracy. This is mainly because the visible light is less susceptible to multipath effects making the propagation and the received optical power more predictable. The best classifier (K-NN) yielded a 99.0% of instances correctly classified and average error distance of only 0.3 centimetres. Also, this classifier was the best performer in terms of precision and recall measurements even for smaller training sets. In addition, the training time spent to build the classifier is the lowest, about 20 milliseconds. On the other hand, the accuracy, precision and recall measurements improve when training dataset size increases, although it needs higher computation effort. Furthermore, the error distance is less than 10 centimetres using only a 60% training dataset size for all classifiers. Hence, it demonstrates that VLC networks may be used for indoor localization based applications with high accuracy constraints.

Since the average error distance of misclassified instances cannot be less than the distance among receivers when classifiers are used, in our ongoing work, we are planning to use other techniques of data mining, such as regression, to reduce the error distance. Moreover, we are also planning to use principal component analysis to reduce the data dimensionality, and hence, the computation time to build the model could be reduced and the system accuracy could be improved.

ACKNOWLEDGEMENTS

This research was partially supported by the Research Program of University of Las Palmas de Gran Canaria (ULPGC2013-15). The fifth author would like to give thanks to the Xunta de Galicia for the financial support given through human resources grant (ED481B 2016/079-0).

| Classifier | Training Size (%) | Accuracy (%) | Error Distance (cm) | Training Time (s) |
|------------------|-------------------|--------------|---------------------|-------------------|
| K-NN | 20 | 76.72 | 10.9 | 0.01 |
| | 40 | 92.13 | 2.9 | 0.01 |
| | 60 | 96.44 | 1.3 | 0.01 |
| | 80 | 98.15 | 0.6 | 0.01 |
| Random Forest | 20 | 63.15 | 34.1 | 2.02 |
| | 40 | 82.94 | 10.7 | 3.85 |
| | 60 | 90.19 | 4.8 | 5.56 |
| | 80 | 93.31 | 2.8 | 7.31 |
| AdaBoost C4.5 | 20 | 57.16 | 41.7 | 1.68 |
| | 40 | 81.28 | 13.9 | 3.46 |
| | 60 | 89.81 | 6 | 5.06 |
| | 80 | 93.42 | 3.4 | 6.55 |
| AdaBoost REPTree | 20 | 23.08 | 84.2 | 0.30 |
| | 40 | 80.85 | 14.8 | 3.11 |
| | 60 | 89.83 | 6.3 | 4.83 |
| | 80 | 93.62 | 3.5 | 6.44 |
| KStar | 20 | 63.75 | 36.6 | 0.01 |
| | 40 | 83.19 | 14.9 | 0.01 |
| | 60 | 91.56 | 6.7 | 0.01 |
| | 80 | 95.54 | 3.2 | 0.01 |
| LMT | 20 | 55.35 | 48.1 | 20.58 |
| | 40 | 79.95 | 20.2 | 58.48 |
| | 60 | 90.15 | 8.8 | 98.70 |
| | 80 | 94.24 | 5.3 | 136.87 |

Table 3. Performance with different training sizes.

REFERENCES

- Armstrong, J., 2013. Visible light positioning: a roadmap for international standardization, *IEEE Communications Magazine*, 51(12), pp. 68-73.
- Aha, D., 1991. Instance-based learning algorithms. *Machine Learning*, 6, pp. 37-66.
- Bahl, P., 2000. RADAR: an in-building RF-based user location and tracking system. In: *IEEE Conference on Computer Communications*, pp. 775-784.

- Breiman, L., 1996. Bagging predictors. *Machine Learning*, 24, pp. 123-140.
- Breiman, L., 2001. Random forests. *Machine learning*, 45, pp. 5-32.
- Cleary, J., 1995. K*: An Instance-based Learner Using an Entropic Distance Measure. In: *12th International Conference on Machine Learning*, Tahoe City, California, USA, pp. 108-114.
- Chen, F., 2008. Performance Evaluation of IEEE 802.15.4 LR-WPAN for Industrial Applications. In: *Fifth Annual Conference on Wireless on Demand Network Systems and Services*, Garmisch-Partenkirchen, Germany, pp. 89-96.
- Chvojka, P., 2015. Channel Characteristics of Visible Light Communications Within Dynamic Indoor Environment. *J. Lightwave Technology*, 33, pp. 1719–1725.
- Cover, T., 1967. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1), pp. 21-27.
- Deqiang, D., 2007. An Optimal Lights Layout Scheme for Visible-Light Communication System. In: *8th International Conference on Electronic Measurement and Instruments*, Xian, China, pp. 2-189 – 2-194.
- Freund, Y., 1996. Experiments with a new boosting algorithm. In: *13th International Conference on Machine Learning*, Bari, Italy, USA, pp. 148-156.
- Hall, M., 2009. The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1), pp. 10-18.
- Honkavirta, V., 2009. A comparative survey of WLAN location fingerprinting methods. In: *Proceedings of the 6th Workshop on Positioning, Navigation and Communication*, Hannover, Germany, pp. 243–251.
- Kaemarungsi, K., 2012. Analysis of WLANs received signal strength indication for indoor location fingerprinting. *Pervasive Mobile Computing*, 8, 292–316.
- Kahn, M., 1997. Wireless Infrared Communications. *Proceedings of the IEEE*, 85(2), pp. 265-298.
- Komine, T., 2004. Fundamental analysis for visible-light communication system using LED lights. *IEEE Transactions on Consumer Electronics*, 50(1), pp. 100–107.
- Landwehr, N., 2005. Logistic Model Trees. *Machine Learning*, 95, pp. 161-205.
- Omnet, 2009. Website of OMNeT++ Discrete Event Simulator. <https://omnetpp.org> (28 April 2009).
- Quinlan R., 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA.
- Srinivasan, B. 2014. Mining Social Networking Data for Classification Using Reptree. *International Journal of Advance Research in Computer Science and Management Studies*, 2(10), pp. 155-160.
- Tronghop, D., 2012. Modeling and analysis of the wireless channel formed by LED angle in visible light communication. In: *International Conference on Information Networking*, Bali, Indonesia, pp. 354-357.
- Want, R., 2001. Expanding the Horizons of Location-Aware Computing. *IEEE Computer*, August 2001, pp. 31-34.