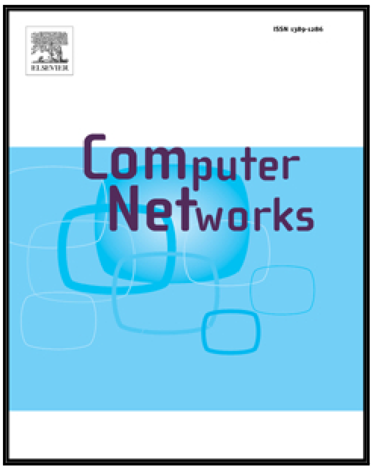

Context-Aware Prediction of Access Points Demand in Wi-Fi Networks

David Rodriguez-Lozano, Juan A. Gomez-Pulido,
Jose M. Lanza-Gutierrez, Arturo Duran-Dominguez,
Ramon A. Fernandez-Diaz

PII: S1389-1286(17)30002-6
DOI: 10.1016/j.comnet.2017.01.002
Reference: COMPNW 6084

To appear in: *Computer Networks*

Received date: 4 August 2016
Revised date: 19 December 2016
Accepted date: 2 January 2017

Context-Aware Prediction of Access Points Demand in Wi-Fi Networks

David Rodriguez-Lozano[a], Juan A. Gomez-Pulido[b,*], Jose M. Lanza-Gutierrez[c], Arturo Duran-Dominguez[d], Ramon A. Fernandez-Diaz[e]

[a] *Computing Service, University of Extremadura, Spain*
[b] *Dep. Technologies of Computers and Communications, University of Extremadura, Spain*
[c] *Center for Industrial Electronics, Technical University of Madrid, Spain*
[d] *Online Campus, University of Extremadura, Spain*
[e] *Dep. Computer and Aerospace Engineering, University of Leon, Spain*

Abstract

We present a methodology based on matrix factorization and gradient descent to predict the number of sessions established in the access points of a Wi-Fi network according to the users' behavior. As the network considered in this work is monitored and controlled by software in order to manage users and resources in real time, we may consider it as a cyber-physical system that interacts with the physical world through access points, whose demands can be predicted according to users' activity. These predictions are useful for relocating or reinforcing some access points according to the changing physical environment. In this work we propose a prediction model based on machine learning techniques, which is validated by comparing the prediction results with real user's activity. Our experiments collected the activity of 1,095 users demanding 26,673 network sessions during one month in a Wi-Fi network composed of 10 access points, and the results are qualitatively valid with regard to the previous knowledge. We can conclude that our proposal is suitable for predicting the demand of sessions in access points when some devices are removed taking into account the usual activity of the network users.

Keywords: Wi-Fi networks, access point, user behavior, prediction, roaming,

*Corresponding author
Email address: `jangomez@unex.es` (Juan A. Gomez-Pulido)

matrix factorization, gradient descent

1. Introduction

Nowadays, Wi-Fi networks are irreplaceable infrastructures in university environments, where students, professors and employees interact with the network for academic and research purposes. They are so important that institutions invest heavily to improve coverage, bandwidth, latencies, security and energy consumptions, among other Quality-of-Service (QoS) parameters.

Wireless infrastructures are usually monitored and controlled by software solutions in order to manage the users and the resources, such as wireless controllers and intelligent Access Points (APs). In this sense, we can consider some aspects of these infrastructures as Cyber-Physical Systems (CPS), since they are composed of physical elements whose operations are supervised by computing and communication elements in smart spaces [1]. Thus, the wireless infrastructure considered in this work interacts with the physical world through hardware devices (the APs), and is managed and audited from a web portal oriented to both hardware resources and users, with the purpose of operating efficiently in real-time.

Optimal placement of APs is a key factor when it comes to planning the network deployment, as these elements allow users to establish network sessions and also transmit the data. An adequate deployment plan is needed because of the hardware features of such devices, that limit their communication capacity and availability.

The physical context of wireless networks understood as communication devices (APs, routers) results in network design efforts aimed at the optimal placement of these elements, trying to maximize coverage and lifetime, as well as to minimize operation costs and energy consumption. However, some aspects were less studied in network deployment, especially those not related to this physical context. Among the possible context-aware aspects, human behavior could have some influence in network improvement and maintenance, since the demand of

APs based on users' habits may suggest locating some elements of the physical context consistently with usual paths, communication strengths, etc.

Once a Wi-Fi network is deployed, and after a lapse of time, those APs with the greatest workloads due to users' demand can be identified. This information is very useful for planning better network operation, as some areas can be reinforced by adding new APs. Besides, knowledge based on the users' behavior is useful when it is necessary to relocate access points due to changing physical environments (spatial restructuring or expansion).

This is the starting point of our research proposal, which is composed of two parts. Firstly, we register the users' behavior in an academic context, *i.e.*, the number of sessions that each user establishes in the APs during a certain time as the main measure of network demand. From this information, prediction models can be built assuming not the simple mathematical fitting, but human behavior. Secondly, these prediction models are applied to some cases of network infrastructure improvement, for example, when the physical environment changes affecting partially relative to the initial AP placement, or instead, when some APs should be reinforced by adding other devices in their proximity, because they support a high workload.

Prediction techniques based on human behavior can be applied for maintenance and to improve the network infrastructure. Hence, for example, we can simulate the network performance when the access point with the highest workload is removed, and predict how the users's activity would move to other APs, as well as the increase (or decrease) of the workloads.

Summarizing, we want to predict the change of access point demand when a specific AP is removed, taking into account the users' behavior. For this purpose, we apply prediction techniques based on Matrix Factorization (MF), which was successfully applied to learning areas.

The remainder of this paper is structured as follows. After going over some related works in Section 2, Section 3 provides the model technique based on matrix factorization for predicting wireless users' performance. In Section 4, we discuss the access points demand prediction, which is validated experimentally in Section 5. Finally, conclusions and future works are left for Section 6.

2. Related Works

There are many aspects of wireless networks worthy of being predicted, especially when the prediction results can be applied to enhance the network infrastructure and users' experience: optimal location, throughput, traffic and mobility, among others. Access points play an important role in these prediction tasks.

The location of communication elements in wireless networks, such as APs, or relay nodes in Wireless Sensor Networks (WSN), is a widely studied research area under the optimization point of view [2]. Following this trend, many parameters involved in the physical context of the wireless networks were optimized, either separately or in combination, trying to maximize coverage [3] and lifetime [4], as well as minimize operation costs (number of devices) and energy consumption [5]. These optimization approaches consider outdoor and indoor environments, as well as many possible constraints, such as, for instance, considering that some locations may be prohibited, or that the number of available resources is limited. Therefore, there are many and assorted optimization approaches, classified as mono-objective or multi-objective optimization problems, and that apply many heuristics and meta-heuristics [6, 7, 8]. Specifically, we can find AP location optimization problems; in this line, [9] solves a problem for indoor WLAN environments using two evolutionary algorithms: Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO). Moreover, users' locations are studied in [10], where a location prediction method based on non-linear time series analysis of the arrival and residence times of users in relevant places is applied to guess their future locations and geographical profiles.

Data throughput and traffic are interesting features to be predicted. Hence, [11] proposes a method that considers terminal distribution to predict the throughput of an AP in a multi-rate, simulated environment. Although traffic demand is usually assumed as static and known a priori, in fact it is really highly dynamic and hardly predictable, even when aggregated at APs. This dynamic and unpredictable nature of wireless traffic demand should be taken into account when applying optimization-based routing solutions. Thus, [12] presents an integrated framework for wireless mesh network routing under dynamic traffic demand, where a traffic estimation method predicts the demand by studying the traces collected at APs using time-series analysis.

Among other network features to be predicted in wireless networks we can find mobility and services. Mobility through a wireless network can be described by the mobile node's next AP. This is the case of [13], that studies prediction agents related to an AP considering real traces of a large Wi-Fi network. In addition, there are cases where the demand of different network elements or services can be predicted for network optimization tasks. Such is the case of [14], that analyzes the application workloads in enterprise environments considering performance modeling and capacity planning, and that predicts future demands based on workload demand patterns in order to build a workload placement recommendation service.

There are many other approaches to human behavior for designing wireless networks. For example, distributed computing in large-scale networked sensor systems are analyzed in the context of human behavior understanding in [15], including a broad range of applications. Closer to our research, the structure of wireless user behavior is characterized in [16] in order to design efficient mobile networks; to this end, this work proposes a similarity metric based on a matrix representation of mobility preferences and its decomposition.

In this work we consider APs as the main elements to predict network sessions according to user behavior. Nevertheless, APs are also interesting for many prediction purposes in the design, deployment, management or maintenance of Wi-Fi networks. For example, estimating the density of nearby APs and the traffic load of the associated Wi-Fi networks [17] facilitate the coexistence between cellular and Wi-Fi networks while sharing the same unlicensed spectrum [18]. On the other hand, studying the users' behavior while interacting with the APs constitutes a new focus of research that allows multiple lines of work. In this line, the data quota that users manage can be included in a redistribution market in order to build an Internet ecosystem where data pricing plays an important role [19]. Other approaches focused on user behavior may be directed to predict the demand for access points, which is our current research interest.

The methodology we followed in our research (modeling, prediction and validation) is similar to that used in other works, where different aspects of the Wi-Fi networks are studied. For example, in [20] a common modeling framework for the number of simultaneously present customers of a nationwide network was developed; by combining statistical methods, this model predicts traffic volumes and patterns, which are compared against test data. Besides statistical methods, learning-based solutions are used for prediction purposes, as in [21], where collaborative filter modeling is applied for portable database workload performance prediction.

3. Prediction Models

Since the activity of the APs is strongly related to the users' behavior, we can apply to our problem those algorithm techniques that have demonstrated good prediction results with systems where the users' preferences and behavior have a great influence. This is the case of the Recommender Systems (RS) [22] and the Predicting Student Performance (PSP) problem [23].

The PSP approach may fit our problem thanks to the similarities of predicting students' performance on particular tasks and predicting wireless network users' performance, considering the usual activity levels of the users when they demand access points in an academic environment. We have adapted the approach of the PSP problem to predict the performance (or activity) p – such as the number of sessions established – of user s at a particular access point i. The mapping of these two prediction problems is shown in Figure 1.

We have chosen the number of sessions as the parameter that best represents users' behavior regarding the use of the network infrastructure. On the contrary, other parameters such as traffic data reflect elements outside of the AP demand,

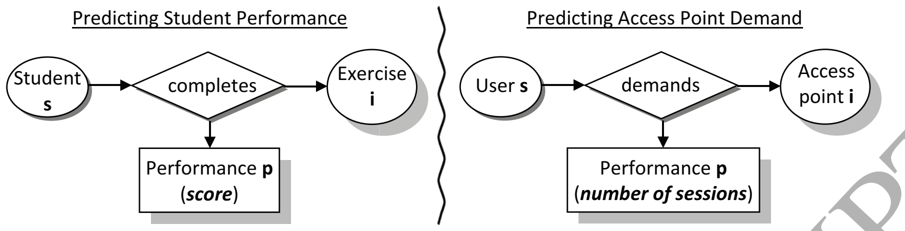

Figure 1: Relationship between PSP and predicting access points demand according to user behavior problems.

as could be those software applications related to social or academic habits of the user, and the mobile technologies that allow different speeds and bandwidths.

The main terms used in the prediction technique are: users set (S), APs set (I), performance values (P), known performances (D^{knw}), unknown performances (D^{unk}), observed or training performances (D^{train}), test performances (D^{test}), and performance predictions ($\hat{P}$).

D^{train} is a subset of D^{knw} used to train the model, which would help predict the unknown performance. The more training values – i.e. observed data – that we use, the better the model we would get. On the other hand, D^{test} are known values chosen to validate the mathematical model using a particular criterion. D^{test} is usually much smaller than D^{train}, and it is used for the performance predictions: $\hat{P} = \hat{p}_1, \hat{p}_2, ..., \hat{p}_{|D^{test}|}$.

3.1. Solving the Prediction Problem by Matrix Factorization

Our goal consists in finding the best model that generates $\hat{P}$. This model can be obtained through Matrix Factorization (MF) [24, 25] if we consider only one relationship between network users and APs, such as "user – demands – access point" (Figure 1). This method is very useful for prediction purposes in recommender systems. Since our mathematical model is very similar to RS, we will use this method to build our prediction proposal.

The prediction model based on matrix factorization considers the number of latent factors, K, which are implicit in the relation "user – demands – access point". This model is implicitly able to encode latent factors of users and APs. The intuition behind using MF is that there should be some latent features that determine how a user interacts with an AP, but it is difficult to establish the proper number of such latent factors. Some suitable factors could be "successful login" and "wrong login", as well as others as "connection time" and "signal strength". Hence, we could make an approximation of the correct number of latent factors studying in depth the wireless environment in relation to the user's context. Anyway, K is not expected to be very high.

Matrix factorization approaches the prediction as a linear combination of factors, which provides good scalability and also allows the use of several algorithm techniques, such as neural networks [26].

This method estimates matrix P as the product of two smaller matrices W_1 and W_2 ($P \approx W_1 W_2^T$) of sizes $S \times K$ and $I \times K$, respectively, where S is the number of users, K is the number of latent factors that describe the user and the access point, and I is the number of APs. Thus,the performance, $p_{s,i}$, that corresponds to user s at access point i is predicted by $\hat{p}_{s,i}$ according to this approach (1).

$$\hat{p}_{s,i} = \sum_{k=1}^{K} (w_{s,k}^{(1)} w_{i,k}^{(2)}) = (W_1 W_2^T)_{s,i} \tag{1}$$

In order to make a prediction model, we first find the best parameters for W_1 and W_2 during the learning phase, that uses D^{train}. Then, the optimal parameters are calculated measuring the differences between real and predicted values using gradient descent. Once the model is obtained, we check its fitness degree using it to predict the values for D^{test}, and measuring the differences with the real values using the Root Mean Squared Error criterion ($RMSE$) (2). Last, the optimal model is used to calculate the unknown values of the performance matrix.

$$RMSE = \sqrt{\frac{\sum_{s,i \in D^{test}} (p_{s,i} - \hat{p}_{s,i})^2}{|D^{test}|}} \tag{2}$$

3.2. Learning Phase

The learning stage finds the optimal values for W_1 and W_2, given a fixed value of K. First, both matrices are initialized with random values – e.g. positive real numbers drawn from the normal distribution $N(0, \sigma^2)$) with standard deviation $\sigma^2 = 0.01$. Then, we calculate the global error (3) from the errors between real and predicted values (4).

$$err = \sum_{(s,i) \in D^{train}} e_{s,i}^2 \tag{3}$$

$$e_{s,i}^2 = (p_{s,i} - \hat{p}_{s,i})^2 + \lambda(||W_1||^2 + ||W_2||^2) \tag{4}$$

The matrix factorization may be an overfit for users with little activity (i.e. few performance): assuming that the vectors of the APs accessed by the user are linearly independent and that W_2 does not change, there exists a vector in W_1 with zero training error. Thus, there is a potential for overfitting, if both the learning rate, β, and the extent of the change in W_2 are small. Regularization factors are often used by machine learning to avoid overfitting [26]. Therefore, $e_{s,i}^2$ includes the regularization factor, λ, that controls the sizes of the factor vectors, so that W_1 and W_2 would give a good approach to P avoiding large values.

The next step minimizes the global error repeatedly updating W_1 and W_2 by means of the Gradient Descent (GD) method [27], which is very efficient with large data sets [28].

In order to apply GD, first, we need to know the gradient of $e_{s,i}^2$ (5)(6) for each value in the dataset, so that we can update $w_{s,k}^{(1)}$ and $w_{i,k}^{(2)}$ in the direction opposite to the gradient (7)(8).

$$\frac{\partial}{\partial w_{s,k}^{(1)}} e_{s,i}^2 = -2e_{s,i} w_{i,k}^{(2)} + \lambda w_{s,k}^{(1)} \tag{5}$$

$$\frac{\partial}{\partial w_{i,k}^{(2)}} e_{s,i}^2 = -2e_{s,i} w_{s,k}^{(1)} + \lambda w_{i,k}^{(2)} \tag{6}$$

$$w_{s,k}^{(1)'} = w_{s,k}^{(1)} - \beta \frac{\partial}{\partial w_{s,k}^{(1)}} e_{s,i}^2 = w_{s,k}^{(1)} + \beta(2e_{s,i} w_{i,k}^{(2)} - \lambda w_{s,k}^{(1)}) \tag{7}$$

$$w_{i,k}^{(2)'} = w_{i,k}^{(2)} - \beta \frac{\partial}{\partial w_{i,k}^{(2)}} e_{s,i}^2 = w_{i,k}^{(2)} + \beta(2e_{s,i} w_{s,k}^{(1)} - \lambda w_{i,k}^{(2)}) \tag{8}$$

W_1 and W_2 are updated repeatedly until some termination criterion is met in order to guarantee, for example, that the error converges to a minimum, or that a preset number of iterations has been accomplished. We have chosen the first option because it provides higher accuracy, although sometimes at the cost of higher computing latencies. Moreover, we have checked that the error converges to a global minimum value, rather than to a local one.

The selection of a good learning rate, β, is a key matter that affects the convergence of the GD algorithm. There are several sophisticated techniques to set that value adaptively [29] [30]. In addition, β may be automatically adjusted between iterations if the algorithm does not converge (i.e. the cost function increases), or in order to accelerate the convergence (i.e. changing β results in a lower value of the cost function).

Finally, the quality of the model is calculated by means of the RMSE criterion on D^{test}, in order to obtain a measure of its fitness degree.

We consider that each experiment is composed of several runs of the same configuration, since the initialization phase contains random values. Hence, for a particular configuration, we can choose the model from the best run. The values of both the learning rate and the regularization term were constant and empirically tuned.

3.3. Prediction Phase

Once W_1 and W_2 are available, the users' performance for the access point i is predicted by (1). The purpose of this prediction is twofold. On the one hand, we can predict unknown values; for example, if an AP has not registered the users' demands during a time. On the other hand, we can recommend access to specific APs according to the users' behavior. From this knowledge, some application possibilities may be explored.

4. Predicting Access Point Demand

We can tackle the problem of predicting the AP's demand according to users' behavior through matrix factorization and gradient descent. If we can predict the number of sessions that the users of the network would establish at the APs, then we will be able to estimate the expected demand at each AP by the accumulation of the expected demands of the users. This information could be very useful for improving the wireless network infrastructure by reinforcing those APs with higher workloads or relocating particular APs, among other possibilities.

4.1. Access Point Infrastructure

The wireless infrastructure considered in our work is based on an open-source distributed solution for managing users and resources of the wireless network (RINUEX) of the University of Extremadura (UEX), in Spain. This architecture was built to satisfy the demand of a large campus disseminated in a wide geographical area (almost 30,000 km^2).

The implementation of this wireless network involves wireless controllers based on open software and intelligent APs, which makes the network independent from technological or commercial aspects, allowing placing the controllers near different campus locations, as well as managing and auditing the network from a web portal, that is oriented to both users and hardware resources.

The current wireless infrastructure is composed of 9 network servers, 686 APs, 28 wireless areas and 68 buildings, involving more than 5,000,000 annual sessions, 20,000 users and 42,000 different devices. For this research, we have considered a local network composed of a few APs placed in several rooms of a library building, and collected data about the number of sessions (understanding them as roamings) that each user establishes at each AP during a determined time frame.

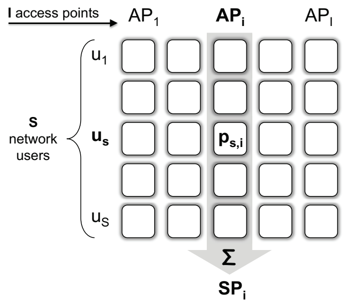

Figure 2: Performance matrix: users, access points and performance (number of sessions), and accumulated performance by access point.

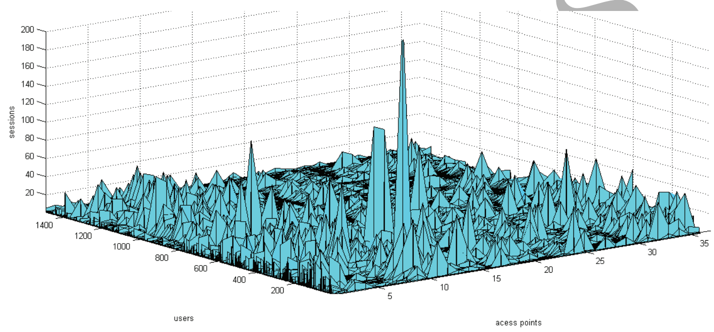

Figure 3: Performance matrix, P.

4.2. Performance Matrix and Experimental Framework

Given a wireless network, we collect data about the number of sessions that each user establishes at each AP, during a determined time frame, and build the corresponding performance matrix P (Fig. 2) of S rows and I columns, where S is the number of network users (u_s) and I is the number of access points (AP_i).

For testing purposes, we considered a data set of 1,517 users and 37 APs of the wireless network at the Polytechnic School of UEX during one month (October 2015). From this data set, we derivea performance matrix, P, with I = 37 rows and S = 1,517 columns, whose values are represented in Figure 3.

It is important to know the current and predicted workloads at each AP according to the users' behavior in order to make easier the analysis and planning

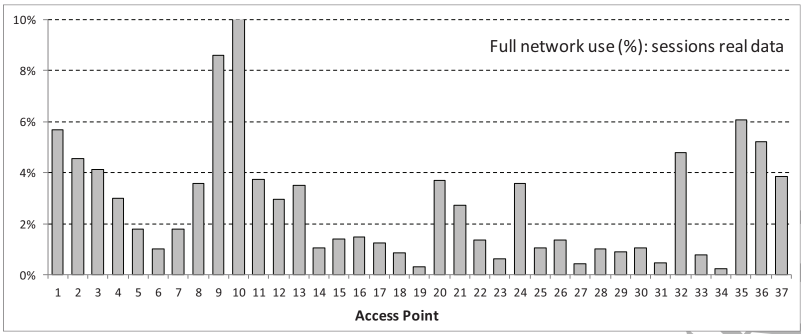

Figure 4: Percentages ofaccumulated performances by access point.

for improving the wireless infrastructure. The workload is understood here as the demand or as thenumber of sessions established by the users. Hence, we calculate the sum, SP_i (Figure 2), of the sessions (9) for each AP_i.

$$SP_i = \frac{\sum_{s=1}^{S} p_{s,i}}{\sum_{i=1}^{I} \sum_{s=1}^{S} p_{s,i}} \times 100\ (\%) \tag{9}$$

There is an important consideration to take into account when using MF: all the values in the performance matrix are known. If a performance value is 0, it does not mean that the data is unknown, but that the user has not established any session at the access point during the considered time.

4.3. Current Network Analysis

We calculate the percentages of the network use from the data collected by the APs as a first evidence of users' behavior. This analysis is useful for finding those APs with the highest and the lowest demands, so that we can compare them with the predicted demand and, then, analyze the impact that removing a particular AP has on the network usage.

Figure 4 shows the percentages of accumulated performance according to (9). We can see that AP_{10} has the highest demand.

4.4. Prediction Model and Parameter Tuning

The prediction model for the wireless network based on users' behavior is built from matrix factorization and gradient descent, using the performance matrix and considering RMSE as the fitness metric. The goal is to predict the network behavior when a specific AP is removed, in order to improve the network through a good deployment or maintenance.

The main parameters of the system prediction model are: number of latent factors, learning rate β, regularization factor λ, and number of runs of each experiment.

The number of latent factors has to do with the factors related to the network context and users' behavior. This number is not known and is difficult to identify, although it might be not very high. We performed a previous experiment where, under the same framework, K was selected from many possible values and the corresponding predictions were calculated. We realized that higher K values caused worse predictions. Therefore, taking into account that the number of possible latent factors should not be very high, we selected K=4 as a reasonable value to be applied in the future experiments.

5. Prediction Model Validation

The prediction method described above has to be validated in order to use it for planning the network infrastructure according to the AP demand based on users' behavior. Pursuing this goal, we follow a methodology composed of three consecutive phases (Figure 5). The first phase ("*collecting data*") begins with the monitoring of the whole network during a determined time in order to collect real data about sessions established by users in APs. Next, we repeat the process during the same time but after removing a particular AP. The second phase ("*prediction*") predicts the behavior for the incomplete network with a performance matrix built using the real data of the entire network, where the performance corresponding to the removed AP are set to zero. Finally, the third phase ("*validation*") validates the quality of the predictions by comparing them with the real performance of the incomplete network.

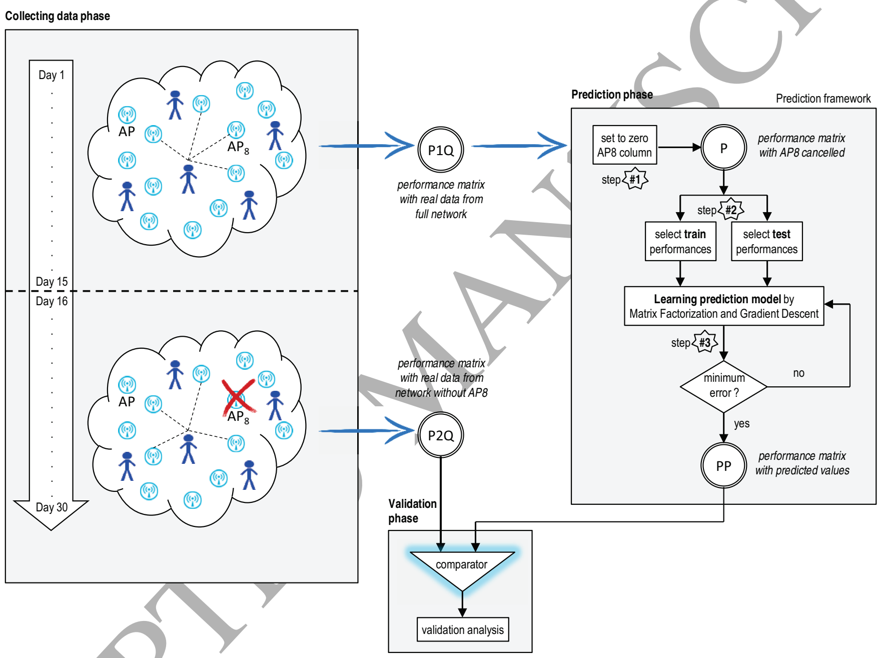

Figure 5: Phases in the prediction model validation.

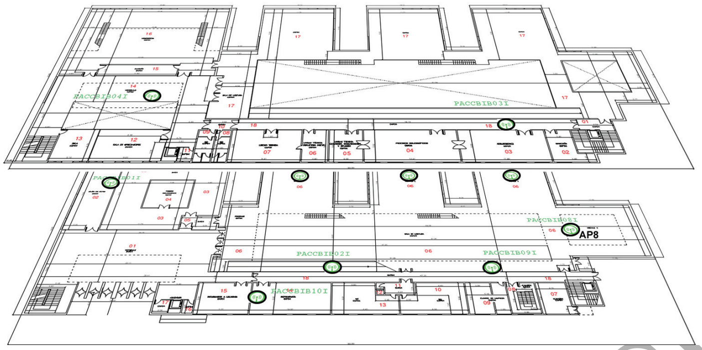

Figure 6: Deployment of 10 access points in the Wi-Fi network of the Library two-story building at the UEX Campus. The access point removed for the empirical validation of the prediction is labeled as AP_8.

5.1. Experimental Framework

We have chosen a Wi-Fi network with high activity: the campus library (Figure 6). This network has 10 access points (AP_1 to AP_{10}) and registers the activity of more than two thousand users a day.

The data collected for the empirical validation of the prediction model represent 1,717 different users during April 2016. This month was split into two halves, where the first half (1Q) keeps the entire network infrastructure untouched, and the second one (2Q) analyzes the same network after switching off one AP (AP_8). This resulted in 1,095 users that showed activity in both of the halves (12,798 and 13,875 sessions, respectively).

5.2. Network Monitoring

Initially, we collect data of the network with all its APs operational during the first half. With these real data we generate the performance matrix $P1Q$. The line $P1Q$ in Figure 7 shows the percentages of the accumulated performances in each AP with regard to the total accumulated performance. In the middle of the month, AP_8 is switched off and the network is monitored again during the second half. Then, the performance matrix $P2Q$ is built, also from real data.

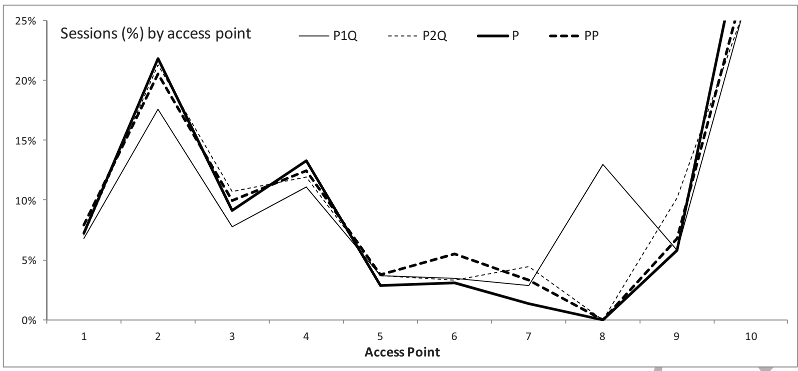

Figure 7: Real and predicted performances for the experimental case $\beta = 0.01$, $\lambda = 0.01$, and $K = 4$.

We can see in Figure 7 that plots $P1Q$ and $P2Q$ are obviously quite different for AP_8, whereas they are similar for the other APs' performance. This tells us that users' behavior pattern is not too different in both halves, because a user often demands network access following similar habits during the month. Nevertheless, the accesses to AP_8 during the first half can influence the remaining APs during the second half (some users of AP_8 could require access now to other APs), hence the increase of demanding some APs in second half can be due to the AP_8 removal.

Our prediction proposal allows us to identify the change of AP use when we measure the difference between the rates of AP demand after removing AP_8. Figure 8 shows the differences of rates between accumulated performance during both the second ($P2Q$) and the first ($P1Q$) halves. These differences show the trend of AP demand when AP_8 is removed. We can see that the demand increases, although in different amounts, for almost all the APs, as former AP_8 users now demand access through other APs. However, this increase may also represent changes in the users' behavior during the second half. Anyway, we can identify the influence of AP_8 removal in the higher demand increases and decreases of the different APs.

Analyzing the plots corresponding to both halves, we can reach some con-

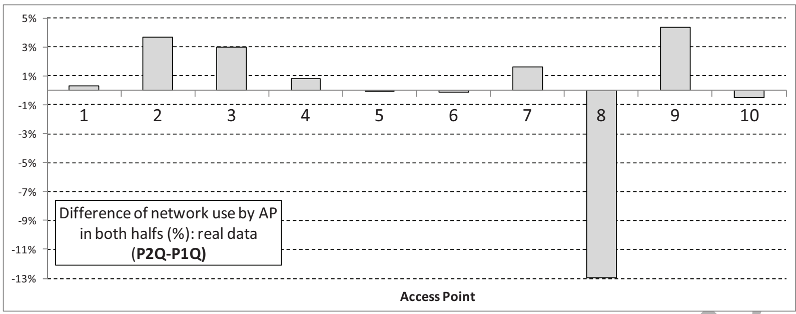

Figure 8: Difference of the percentages of accumulated real performances between the second ($P2Q$) and the first half ($P1Q$), showing the trends of APs demand when removing AP_8.

Table 1: Real demand variation of the APs after removing AP_8.

APs with increases:	(x > 2%)	AP_2, AP_3, AP_9
APs almost unaltered:	(-2% < x < 2%)	AP_1, AP_4, AP_5, AP_6, AP_7, AP_{10}
APs with decreases:	(-2% < x)	AP_8

clusions about the behavior of real users when AP_8 is removed. Figure 8 shows that the demand changes at some APs after removing AP_8. If we would not have removed AP_8 in the second half and all the users demanded the same APs with unaltered habits, all the columns in the graph would have had zero heights. Nevertheless, those users that demanded AP_8 during the first half would have to demand other APs after removing it, and that is represented by higher columns in the figure. This way, column heights in Figure 8 inevitably reflect the human behavior in both halves with regard to the transfer of users from AP_8 to other APs. Certainly, this is not a conclusion, but a valid assumption.

Trying to find the reason for the increased height in each column, we can establish the following reasonable hypothesis for analyzing the data: if users show similar behaviors in both halves, the greatest part of the variation in the APs demand is due to AP_8 removal. In addition, the longer the time we collect data, the more similar the behavior will be. According to this hypothesis, we identify the more significant variations in the APs in Figure 8, and classify them as shown in Table 1.

This study was accomplished using a real framework with and without AP_8 (both halves), but we want to find out whether it is possible to predict variations such as those shown in Table 1 from only the full network real data (first half).

5.3. Network Prediction

Our prediction proposal consists of three steps, as Figure 5 shows in the "Prediction framework" box. First (step #1), we consider the AP removal in the performance matrix $P1Q$ by setting to zero the corresponding column, generating the performance matrix P, needed for the prediction. Hence, once the performances corresponding to AP_8 are set to zero, the rates of AP demand are recalculated; this is the reason why plot P is slightly different from plot $P1Q$ in Figure 7 (obviously, they meet at access point AP_8).

The second step selects data from P, which would be considered as "unknown" performance for the predictions. These data are selected by choosing one performance value for each user in the matrix, covering the different APs consecutively. This method tries to obtain sufficient representative performance to achieve the prediction, allowing P to keep enough real performance values to retain a certain "memory" of the users' behavior pattern.

The third step applies matrix factorization to P and generates the prediction matrix PP. This matrix shares the same values as P, except those corresponding with the unknown data set, which are replaced by the performances predicted through the matrix factorization model. This way, the goal is to make PP similar to $P2Q$; in other words, PP tries to predict $P2QS$ from the information provided by $P1Q$ (which is always available). Note that $P2QS$ contains unknown values when we apply this prediction methodology to any problem, but that those values are known for the empirical validation. We can see in Figure 7 that the plot line that represents PP is quite similar to $P2Q$. This plot has been generated for an experimental case with $K = 4$, $\lambda = 0.01$, $\beta = 0.01$, and 1,000 iterations. Obviously, different values of the parameters would result in different plots of PP.

5.4. Prediction Validation

In order to validate the prediction, we use a method for qualitative confirmation and consider a metric for parameter tuning. Qualitative confirmation tries to measure the difference of the accumulated performances PP-$P1Q$ and check if the increase/decrease trends of the APs demand match the corresponding trends for the difference $P2Q$-$P1Q$ (shown by Figure 8).

Prediction results depend on the correct selection of the main parameters (β and λ), so a metric for tuning them is needed to guarantee optimal predictions. The best values would be those that minimize the difference between predicted performance (PP) and real performance in the second half ($P2Q$). Thus, the set of values that minimize the cost function $Fval$ (10) is considered as the optimal set, this function being defined as the absolute difference between the accumulated performances SPP (11) and $SP2Q$ (12), by AP.

$$Fval = |\sum_{i=1}^{I} SPP_i - \sum_{i=1}^{I} SP2Q_i| \tag{10}$$

$$SPP_i = \frac{\sum_{s=1}^{S} pp_{s,i}}{\sum_{i=1}^{I}\sum_{s=1}^{S} pp_{s,i}} \times 100\ (\%) \tag{11}$$

$$SP2Q_i = \frac{\sum_{s=1}^{S} p2q_{s,i}}{\sum_{i=1}^{I}\sum_{s=1}^{S} p2q_{s,i}} \times 100\ (\%) \tag{12}$$

Since the learning rate is crucial for the convergence, and the regularization factor affects the accuracy of $Fval$, we have tackled a set of prediction experiments, where $Fval$ was calculated considering the same set of values for both β and λ: 100, 75, 50, 25, 10, 5, 2.5, 1, 0.5, 0.1, 0.05, 0.01, and 0.005. Based on previous experiments, we considered $K = 4$ latent factors and 100 runs for each experiment, and found out that $\beta = 1$ and $\lambda = 75$ minimize $Fval$ (see Figure 9).

After selecting theoptimal values, we analyze the predictions in order to qualitatively validate the trend of APs demand. Hence, we plot the difference of the accumulated performances, PP-$P1Q$, and check whether this trend is

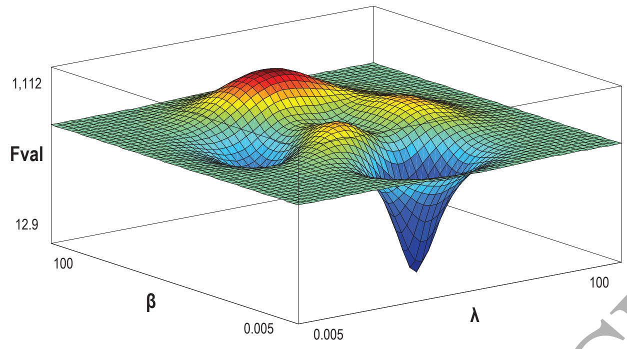

Figure 9: Graphical view of Fval values obtained from predictions for different β and λ values. There is a global minimum (12.923) corresponding with $\beta = 1$ and $\lambda = 75$.

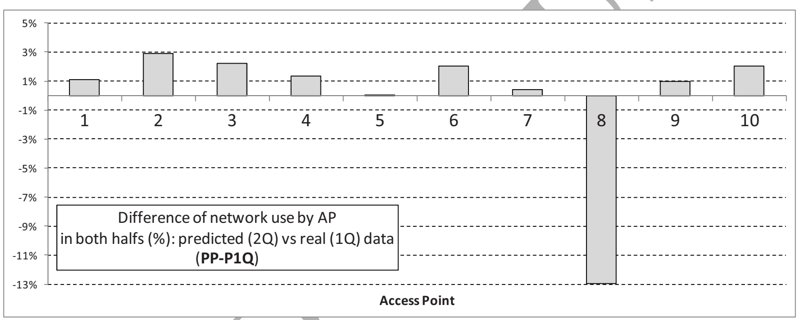

Figure 10: Difference of the percentages of accumulated performances between behavior predicted in the second half (PP) and real behavior in the first half ($P1Q$), showing the predicted trends of AP demand when removing AP_8.

similar to the $P2QS$-$P1QS$ trend shown by Figure 8. Thus, Figure 10 shows the comparison between predicted data when AP_8 was removed, and real data from the whole network. From this figure we have built Table 2 in order to analyze the trend after removing AP_8.

Comparing both Table 1 and Table 2, we can conclude that the prediction is close to the real behavior. The prediction has identified the two access points with the highest increase of demand (AP_2 and AP_3) as well as a very similar decrease for AP_8. The remaining APs present short differences with regard to the real behavior. These results support the idea that a prediction method

Table 2: Predicted demand variation of APs after removing AP_8.

APs with increases:	(x > 2%)	AP_2, AP_3
APs almost unaltered:	(-2% < x < 2%)	AP_1, AP_4, AP_5, AP_6, AP_7, AP_9, AP_{10}
APs with decreases:	(-2% < x)	AP_8

based on matrix factorization and gradient descent can be successfully applied to the analysis of access point demand according to user's behavior. Moreover, the results in the prediction validation have shown an additional detail: the existence of a global minimum when searching for the best values of the couple (β, λ). Applying the learning phase to the model means finding the optimal matrices W_1 and W_2 (solutions) for a particular value of K, once the values of β and λ have been chosen. After performing an exhaustive search experiment where β and λ were selected from a wide range, we checked that the optimal solution is a global optimal too. This result is important because it opens a new research line where optimization algorithms based on evolutionary computing could be used to speed-up the search for optimal solutions.

6. Conclusions and Future Works

This work explores the application of prediction techniques based on matrix factorization and gradient descent to determine the access point demands in Wi-Fi infrastructures according to users' behavior. From real and predicted data in a university environment, the results of the experiments indicate, from a qualitative point of view, that it is possible to obtain consistent, valid approaches to be applied to wireless networks taking into account the users' preferences in their usual activities. For example, the access point workload prediction could be used in order to achieve a more efficient deployment (placing the APs where the users can get more benefit), or to maintain the wireless network infrastructure by adding or reinforcing the corresponding hardware devices.

Another research line should take into account larger time frames for collecting data, because of the stochastic nature of user behavior. In addition, collecting data during different time periods would allow analyzing the impact of time domain in the prediction. Finally, we would like to explore self-tuning methods for the learning rate and regularization factor in order to improve the prediction accuracy.

References

[1] R. Rajkumar, I. Lee, L. Sha, J. Stankovic, Cyber-physical systems: The next computing revolution, in: Proceedings of the 47th ACM/IEEE Design Automation Conference (DAC), IEEE, Anaheim, CA, USA, 2010, pp. 731–736.

[2] M. Younis, K. Akkaya, Strategies and techniques for node placement in wireless sensor networks - a survey, Ad Hoc Networks 6 (2008) 621–655.

[3] S. Indu, A. Bhattacharyya, Self deployment of mobile sensor network for optimal coverage, International Journal of Engineering Science and Technology 2 (7) (2010) 2968–2975.

[4] S. Basagni, A. Carosi, C. Petrioli, Heuristics for lifetime maximization in wireless sensor networks with multiple mobile sinks, in: Proceedings of the 2009 IEEE International Conference on Communications, 2009, pp. 1–6.

[5] A. Konstantinidis, K. Yang, Multi-objective k-connected deployment and power assignment in wsns using a problem-specific constrained evolutionary algorithm based on decomposition, Computer Communications 34 (2011) 83–98.

[6] P. Boonma, J. Suzuki, Evolutionary constraint-based multiobjective adaptation for self-organizing wireless sensor networks, in: Proc. of the 2nd ACM/IEEE/Create-Net/ICST International Conference on Bio-Inspired Models of Network, Information and Computing Systems (BIONETICS), Budapest, Hundary, 2007, pp. 111–119.

[7] N. Banerjee, R. Kumar, Multiobjective network design for realistic traffic models, in: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, London, England, 2007, pp. 1904–1911.

[8] J. M. Lanza-Gutierrez, J. A. Gomez-Pulido, Assuming multiobjective metaheuristics to solve a three-objective optimisation problem for relay node deployment in wireless sensor networks, Applied Soft Computing 30 (2015) 675–687.

[9] I. Vilovic, N. Burum, Location optimization of wlan access points based on a neural network model and evolutionary algorithms, Journal for Control, Measurement, Electronics, Computing and Communications 55 (3) (2014) 317–329.

[10] S. Scellato, M. Musolesi, C. Mascolo, V. Latora, A. T. Campbell, NextPlace: A Spatio-temporal Prediction Framework for Pervasive Systems, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 152–169. doi:10.1007/978-3-642-21726-5_10.

[11] R. Hamamoto, C. Takano, H. Obata, K. Ishida, Improvement of throughput prediction method for access point in multi-rate wlans considering media access control and frame collision, in: 2015 Third International Symposium on Computing and Networking (CANDAR), 2015, pp. 227–233. doi:10.1109/CANDAR.2015.79.

[12] L. Dai, Y. Xue, B. Chang, Y. Cao, Y. Cui, Integrating traffic estimation and routing optimization for multi-radio multi-channel wireless mesh networks, in: INFOCOM 2008. The 27th Conference on Computer Communications. IEEE, 2008. doi:10.1109/INFOCOM.2008.23.

[13] J.-M. François, G. Leduc, AP and MN-Centric Mobility Prediction: A Comparative Study Based on Wireless Traces, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 322–332. doi:10.1007/978-3-540-72606-7_28.

[14] D. Gmach, J. Rolia, L. Cherkasova, A. Kemper, Workload analysis and demand prediction of enterprise data center applications, in: Proceedings of the 2007 IEEE 10th International Symposium on Workload Characterization, IISWC '07, IEEE Computer Society, Washington, DC, USA, 2007, pp. 171–180. doi:10.1109/IISWC.2007.4362193.

[15] P. Spagnolo, P. Mazzeo, C. Distante (Eds.), Human Behavior Understanding in Networked Sensing, Springer, 2014.

[16] W. J. Hsu, D. Dutta, A. Helmy, Structural analysis of user association patterns in university campus wireless lans, IEEE Transactions on Mobile Computing 11 (11) (2012) 1734–1748. doi:10.1109/TMC.2011.254.

[17] H. Zhang, X. Chu, W. Guo, S. Wang, Coexistence of wi-fi and heterogeneous small cell networks sharing unlicensed spectrum, IEEE Communications Magazine 53 (3) (2015) 158–164. doi:10.1109/MCOM.2015.7060498.

[18] H. He, H. Shan, A. Huang, L. X. Cai, T. Quek, Proportional fairness-based resource allocation for lte-u coexisting with wi-fi, IEEE Access PP (99) (2016) 1–1. doi:10.1109/ACCESS.2016.2604822.

[19] Y. Shen, C. Jiang, T. Q. S. Quek, H. Zhang, Y. Ren, Pricing equilibrium for data redistribution market in wireless networks with matching methodology, in: 2015 IEEE International Conference on Communications (ICC), 2015, pp. 3051–3056. doi:10.1109/ICC.2015.7248792.

[20] A. Ghosh, R. Jana, V. Ramaswami, J. Rowland, N. K. Shankaranarayanan, Modeling and characterization of large-scale wi-fi traffic in public hotspots., in: INFOCOM, IEEE, 2011, pp. 2921–2929.

[21] J. Duggan, Y. Chi, H. Hacigumus, S. Zhu, U. Cetintemel, Packing light: Portable workload performance prediction for the cloud, 2013, pp. 258–265. doi:10.1109/ICDEW.2013.6547460.

[22] D. Jannach, M. Z. amd A. Felfernig, G. Friedrich, Recommender Systems. An Introduction, Cambridge University Press, 2011.

[23] N. Thai-Nghe, L. Drumond, T. Horvath, A. Krohn-Grimberghe, A. Nanopoulos, L. Schmidt-Thieme, Factorization techniques for predicting student performance, in: Educational Recommender Systems and Technologies: Practices and Challenges, IGI-Global, 2012, pp. 129–153.

[24] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, Computer 42 (8) (2009) 30–37.

[25] S. Rendle, L. Schmidt-Thieme, Online-updating regularized kernel matrix factorization models for large-scale recommender systems, in: Proceedings of the 2008 ACM conference on Recommender systems, 2008, pp. 251–258.

[26] G. Takacs, I. Pilaszy, B. Nemeth, D. Tikk, Scalable collaborative filtering approaches for large recommender systems, The Journal of Machine Learning Research 10 (2009) 623–656.

[27] Y. Koren, Factor in the neighbors: Scalable and accurate collaborative filtering, ACM Transactions on Knowledge Discovery from Data 4 (2010) 1–24.

[28] L. Bottou, Large-scale machine learning with stochastic gradient descent, in: Proceedings of 19th International Conference on Computational Statistics, Springer, 2010, pp. 177–186.

[29] R. Zhang, W. Gong, V. Grzeda, A. Yaworski, M. Greenspan, An adaptive learning rate method for improving adaptability of background models, IEEE Signal Processing Letters 20 (12) (2013) 1266–1269.

[30] R. Ranganath, C. Wang, D. M. Blei, E. P. Xing, An adaptive learning rate for stochastic variational inference, in: Proceedings of 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013, pp. 1–9.

David Rodriguez-Lozano received the Ms.C. degree in computer science from the University of Extremadura, Caceres, Spain, in 1994. He is currently assistant professor of digital control in the Department of Technology of Computers and Communications, University of Extremadura, Spain, and head of the Computing Service of this universsity, where he is in charge of the institutional wireless infraestructure. He has authored or co-authored 8 publications and 11 peer-reviewed conference proceedings, and developed research works in web technologies, Wi-Fi networks, logic design, VLSI circuits and reconfigurable architectures based on FPGA devices. His current research interest falls mainly on developing prediction techniques for enhancing wireless infraestructures according to human context.

Juan A. Gomez-Pulido received the Ph.D. degree in physics, electronics specialty, from the Complutense University, Madrid, Spain, in 1993. He is currently professor of computer organization and design of processors in the Department of Technology of Computers and Communications, University of Extremadura, Spain. He has authored or co-authored 60 ISI journals, tens of book chapters, and more than two hundred peer-reviewed conference proceedings. His current research interests fall within hot topics on wireless sensor networks, mobile computing, machine learning applied to big-data analysis, reconfigurable and embedded computing based on FPGA devices, optimization, and evolutionary computing.

Jose M. Lanza-Gutierrez received the Ph.D. degree in computer science from the University of Extremadura, Caceres, Spain, in 2015. He is currently a postdoc researcher at Center of Industrial Electronics, Universidad Politecnica de Madrid, Madrid, Spain. He has authored or co-authored more than 30 publications, including ISI journals, book chapters, and peer-reviewed conference proceedings. His current research interests fall on machine learning, big data, metaheuristics, and parallel and distributed computing.

Arturo Duran-Dominguez received the Ms.C. degree in computer science from the University of Extremadura, Caceres, Spain, in 2001. He is currently assistant professor of computer organization in the Department of Technology of Computers and Communications, University of Extremadura, Spain. He develops professional activities in the Online Campus of this university, in a strong relationship with his current research interests: big data analysis from e-learning tools.

Ramon A. Fernandez-Diaz received the PhD degree in computer sciences from the University of Leon, Spain, in 2001. He is currently a professor of Computer Architecture in the Department of Computer Engineering, University of Leon, Spain. His current research interests are intelligent heuristics and computer architectures for solving optimization problems.