

Article

Proactive Cross-Layer Framework Based on Classification Techniques for Handover Decision on WLAN Environments

Josué Vicente Cervantes-Bazán ¹, Alma Delia Cuevas-Rasgado ², Luis Martín Rojas-Cárdenas ³,
Saúl Lazcano-Salas ⁴, Farid García-Lamont ², Luis Arturo Soriano ¹, José de Jesús Rubio ^{5,*}
and Jaime Pacheco ⁵

¹ Departamento de Ingeniería Mecánica Agrícola, Universidad Autónoma Chapingo, Texcoco, Km. 38.5 Carretera México-Texcoco, Chapingo, Texcoco 56230, Mexico; jcervantesba@chapingo.mx (J.V.C.-B.); larturo.soriano@gmail.com (L.A.S.)

² Computer Engineering, Universidad Autónoma del Estado de México CU Texcoco, Jardín Zumpango Str., Texcoco 56259, Mexico; adcuevasr@uaemex.mx (A.D.C.-R.); fgarcial@uaemex.mx (F.G.-L.)

³ Electronic Engineering, Universidad Autónoma Metropolitana Iztapalapa, San Rafael Atlixco Str., 186, Iztapalapa, Ciudad de México, Mexico City 09340, Mexico; lmrc@xanum.uam.mx

⁴ Computer Engineering, Universidad Autónoma del Estado de México CU Valle de México, University Boulevard, La Paz 56500, Mexico; slazcanos@uaemex.mx

⁵ Sección de Estudios de Posgrado e Investigación, ESIME Azcapotzalco, Instituto Politécnico Nacional, Av. de las Granjas no. 682, Col. Santa Catarina, Ciudad de México, Mexico City 02250, Mexico; jpachecom@gmail.com

* Correspondence: Correspondence: rubio.josedejesus@gmail.com or jrubioa@ipn.mx; Tel.: +52-55-5729-6000 (ext. 64497)



Citation: Cervantes-Bazán, J.V.; Cuevas-Rasgado, A.D.; Rojas-Cárdenas, L.M.; Lazcano-Salas, S.; García-Lamont, F.; Soriano, L.A.; Rubio, J.d.J.; Pacheco, J. Proactive Cross-Layer Framework Based on Classification Techniques for Handover Decision on WLAN Environments. *Electronics* **2022**, *11*, 712. <https://doi.org/10.3390/electronics11050712>

Academic Editors: Imad Rida, Lunke Fei and Dan Istrate

Received: 31 December 2021

Accepted: 23 February 2022

Published: 25 February 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Abstract: In recent years, modern technology has been increasing, and this has grown a derivative in big challenges related to the network and application infrastructures. New devices have been providing more high functionalities to users than ever before; however, these devices depend on a high functionality of network in order to ensure a correct functioning ability over applications. This is essential for mobile networking systems to evolve in order to meet the future requirements of capacity, coverage, and data rate. In addition, when a network problem happens, it could be converted into something more disastrous and difficult to solve. A crucial point is the network physical change and the difficulties, such as loss continuity of services and the decision to select the future network to be connected. In this article, a new framework is proposed to forecast a future network to be connected through a mobile node in WLAN environments. The proposed framework considers a decision-making process based on five classifiers and the user's position and acceleration data in order to anticipate the network change, reaching up to 96.75% accuracy in predicting the connection of this future network. In this way, an early change of network is obtained without packet and time loss during the network change.

Keywords: cross-layer; handover; handoff decision; naive bayes; logistic regression; decision tree; k-nearest neighbors; support-vector machines

1. Introduction

Mobile devices have acquired greater capacity for computing, storage, and connectivity to heterogeneous networks. Nevertheless, a critical point of mobility is the network physical change, and the problems inherent to the loss of continuity of services, as well as the decision to change, that is, select the future network to be connected [1].

The handover or handoff in IP networks is the physical transition from one network to another. The handover is typified by two types of transition: (A) horizontal handover and (B) vertical handover [2]. When a mobile node (MN) changes network in the same technology, it performs a horizontal handoff. However, if the change of network is to a different kind of technology, then it is called a vertical handoff.

The general procedure of change is classified into three stages according to [3,4]:

- (1) Measurement of handover and initialization.
- (2) Decision of handover.
- (3) Execution of handover.

In the first stage, the mobile node (MN) takes measurements of the metrics of the nearby networks. In the second stage, algorithms decide when and which networks to change. Finally, in the third stage, the necessary procedures are made to connect to the new network and reestablish services.

Regarding problems related to the execution of the handover, IP establishes that for all equipment that operates within a network, its address must be derived from this network. Under this scheme, if a computer moves from its original network to a new network, then it will experience the following problems:

- (1) All communication becomes impossible because its IP address is not valid in the new network.
- (2) Communications in progress are lost.
- (3) Mobile nodes disappear from the global network.

The problems related to the handover decision arise in the design of an algorithm that determines the right moment to change networks, because it is possible that the decision-making algorithm is forced by the MN to switch successively between adjacent networks. Although the MN is not in movement, this is because the change of network algorithm is based on a measurement of the received signal strength (RSS) and the MN has a very similar received signal strength indicator (RSSI) between both adjacent networks.

This is known as the ping-pong effect [5]. This is mainly generated when the decision-making algorithm is based on selecting the network with the highest received signal strength (RSSI). Thus, there are two networks that have a similar power, and the algorithm makes the change to the strongest signal recurrently. This ping-pong effect is a very common phenomenon that causes degradation in the quality of services. Besides, the ping-pong effect is very common not only in network exchange, but it occurs in various areas, and a very common strategy to combat it is to establish limits by ranges. For example, if the frontier is 6, it does not go to the next level until it has a power of 6.5 or higher, and it does not return to the lower level until it is 5.5 or lower. In this way, this border around 6 is blurred, and it reduces that ping pong effect. Another wall is through algorithms such as fuzzy logic [6–9], where the idea is very similar. In other words, they use these policies to make change decisions and at the same time try to balance the load of the networks efficiently, using input parameters such as RSSI, latency, and data rate.

Generally speaking, the decision-making algorithm is maintained with the data provided by the network and after processing the data, and it decides which network to switch to. They are usually changing between the wireless local area network (WLAN), universal mobile telecommunications system (UMTS), long term evolution (LTE), among others. Multiple proposals use artificial intelligence techniques, such as artificial neural networks (ANN), fuzzy logic (FL), genetic algorithms (GA), decision trees (DT), among others [10–13]. In [14], the authors proposed an algorithm which optimizes the feature boundary of deep convolutional neural networks (CNN) in order to reduce the overfitting problem, and this is a strategy to deal with the two-stage training process. In [15], the authors proposed an algorithm to estimate the depth map and faces from monocular 2D images. This algorithm helps to deal with the difficulties such as accessibility, cost, and privacy of data. In [16], an algorithm is formulated and evaluated to determine the facial diagnosis which uses and a small dataset. Similarly, [17] gives an overview related to reinforcement learning algorithm for handover management in 5G ultra-dense small cell (UDSC) Networks. In this study, a variety of artificial intelligence algorithms is presented, and the authors give future directions and challenges for 5G UDSC networks. In [18], a data augmentation framework is formulated to improve the accuracy of deep convolutional neural networks that helps to reduce the training cost and in the same way improve its generalization ability. A profound

analysis of decision-making algorithms is carried out in the related works section, although there are few works that consider the prognosis of MN movement.

This research focuses on proposing a framework to reduce packet loss during handover, forecasting the movement of the mobile node using classification techniques. Therefore, to anticipate the network processes, it is necessary to carry out the handover, generating a pro-active proposal and avoiding packet loss in mobile WLAN environments. Thus, the main contribution of this framework proposed is maintained the communication on track by anticipating the network switch in order to ensure continuity of services.

The main challenge is to delimit the area to train the classifiers, i.e., a lot of data generates a longer training time, and it is possible that the model will never fit. This challenge will be analyzed in future research.

This work is organized as follows: in Section 2, a state-of-the-art review of related works is presented, in Section 3 the problem of selecting the next network to connect a mobile device is described, in Section 4 a proactive cross-layer framework design based on artificial intelligence techniques for seamless handover in mobile WLAN environments is proposed and formulated. In Section 5, the results are presented and discussed. Finally, the conclusions are given.

2. Related Works

The different proposals can be found divided into two areas: (1) The proposals to improve handover protocols and (2) The decision-making proposals.

In the first section, you can find protocols that focus on reducing the scanning time such as [19–21]; on the other hand, there are proposals that focus on reducing the handover time such as [22,23]. This work focuses on improving decision-making, that is, in point 2.

Neural networks in the handover have been applied to solve decision-making changes in heterogeneous networks. In [24], the authors propose a back-propagation artificial neural network (ANN) using the RSS input parameters and the traffic intensity in the target networks, monitoring the training of the network. However, the delay caused by the training stage is a problem. In [25], a middleware based on an ANN is proposed to select the best network based on user preferences. Furthermore, it increases latency during handover execution due to the size of the signaling packets used and the time of training. Another proposal based on the use of RN is made in [26] where a neural network with RSSI input parameters and the speed of the MN is proposed, reducing the number of unnecessary handovers.

Vertical handover algorithms involve several factors that can make difficult the quantification. Fuzzy logic can be applied to solve change decision problems like [7–9], [27] that use these policies to make decision changes, and at the same time they try to balance the load of the networks efficiently, using input parameters such as RSSI, latency, and data rate.

The trend in reviewed literature shows that the networks will be more heterogeneous, and in this way, a vertical handover will be more normal. Some of the reviewed algorithms are classified based on the main criteria used to make the decision.

Regarding the decision-making proposals, multiple algorithms have been proposed. These use schemes based on received signal strength (RSS), quality of service (QoS), decision functions based on multi-criteria, and algorithms based on artificial intelligence techniques. In general terms, the decision-making algorithm feeds on the data provided by the network. After processing the data, it decides which network to switch to. Some of the criteria commonly used to make this decision to change are illustrated in the following topic: network parameters.

Network Parameters

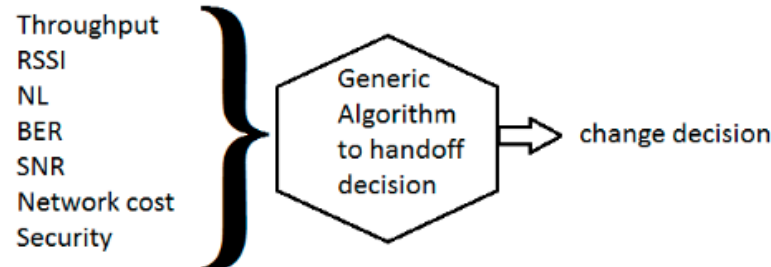
The network parameters or criteria frequently used in handover decision-making are mentioned below, along with their description in Table 1.

Table 1. Features commonly used in handover decision algorithms.

Parameter	Description
Received signal strength indicator (RSSI)	Power level of received signals in wireless networks, measure on decibels (dB).
Network load (NL)	Loads of traffic on a network can be an important parameter due to the capacity of the channel.
Bit error rate (BER)	Number of received bits that have been altered due to noise or interference, divided by the total number of bits transferred during the interval time, measure on bits per second (bps).
Throughput	This measure refers to the amount of data or messages that are successfully received, on a specific channel, measure on bits per second (bps).
signal to noise ratio (SNR)	It is the relation between the signal to noise and is defined as the ratio between the transmitted signal power and the power of the noise that corrupts it, measure on decibels (dB).

In addition, some user preferences such as the cost of the network and security are considered.

In Figure 1, a generic algorithm to the handover decision phase is shown, which is basically a representation of the input data that is passed to the algorithm in order to make a change decision [19], [28–32]. That is, select the next network. This work focuses on the movement of the mobile node to predict the next network to connect. For this reason, this article proposes parameters such as position, speed, acceleration, and power of the received signal (RSS) as features.

**Figure 1.** Generic algorithm to the handover making decision.

The algorithms that make the decision to change based on artificial intelligence are associated with high algorithmic complexity. Therefore, in this paper, different criteria are proposed to perform a network change's forecast based on classification techniques. In this context, the next sub-section presents the problem to be solved after the simulation of the proactive cross-layer protocol based on artificial intelligence techniques for seamless handover in mobile WLAN environments.

3. The Problem of Selecting the Next Network

The problem of selecting the next network can be formulated as three sets as follows: given a set R of candidate networks to carry out the handover, consisting of the triple $\{\text{Extended Service Set Identifier (ESSID)}, \text{Received Signal Strength Indicator (RSSI)}, \text{Mac Address (MA)}\}$, that is, let $R = \{(ESSID_1, RSSI_1, MA_1), \dots, (ESSID_{|R|}, RSSI_{|R|}, MA_{|R|})\}$, $|R| \geq 2$, and another GPS set formed by the position of the MN in decimal degrees constituted by the triplet $\{\text{latitude}(\text{lat}), \text{longitude}(\text{lon}), \text{height}(\text{alt})\}$. That is, let $GPS = \{(\text{lat}_1, \text{lon}_1, \text{alt}_1), \dots, (\text{lat}_{|N|}, \text{lon}_{|N|}, \text{alt}_{|N|})\}$, $|N| \geq 2$, and finally the set AV consisting of the acceleration in the three axes, x -axis(ax), y -axis(ay), and z -axis(az), and the velocity (v) $AV = \{(ax_1, ay_1, az_1, v_1), \dots, (ax_{|A|}, ay_{|A|}, az_{|A|}, v_{|A|})\}$, where $|A| \geq 2$. The solution is modeled with the set $D_i^t = \{GPS^t, AV^t, R_i^t\}$, $i = 1, \dots, R$ and $t > 0$.

Then the data set is expressed as follows:

$$D_i^t = \{\text{lat}, \text{lon}, \text{alt}, \text{ax}, \text{ay}, \text{az}, \text{v}, \text{ESSID}, \text{RSSI}, \text{MA}\}$$

That is, at time $t = 1$, there are three networks available to perform the network change.

$$D_3^1 = \{19.41356, -98.90195, 2242, 1.41617, 3.84269, 7.96311, 5.01, \text{Net1}, -78, \text{e4:3e:d7:26:cd:a7}, \text{Net2}, -78, 48:8d:36:3d:04:b8, \text{Net3}, -85, 94:8f:cf:84:67:32\}$$

Moreover, at time $t = 2$, there are two networks available to perform the network change.

$$D_2^2 = \{19.491490, -98.892851, 2267, 5.75562, -0.153, 0.306, 10.726, \text{INFINITUM68xx}, -78, 50:4e:dc:2b:fe:18, \text{INFINITUMr5xz}, -78, 7c:b1:5d:5e:4e:f8\}$$

The data set D is variable depending on the coverage of closed networks to the position. It is important to mention that in the trivial case where there is only one network available to change, by definition that is the best and worst network at the same time. Therefore, the decision-making algorithm can only evaluate one possibility and it would be the only option to select. For these reasons, in this article an environment with at least two or more adjacent networks is considered.

4. Proactive Cross-Layer Framework Design Based on Artificial Intelligence Techniques for Seamless Handover in Mobile WLAN Environments

This section explains the design of the test environment in the proactive cross-layer framework. It is assumed that the MN is in movement, and it is in an area fully covered with multiple WLAN networks as shown in Figure 2. The main objective of the protocol is to forecast the next network the MN will connect to. In this scenario, eight access points were placed on a path as shown in Figure 2.

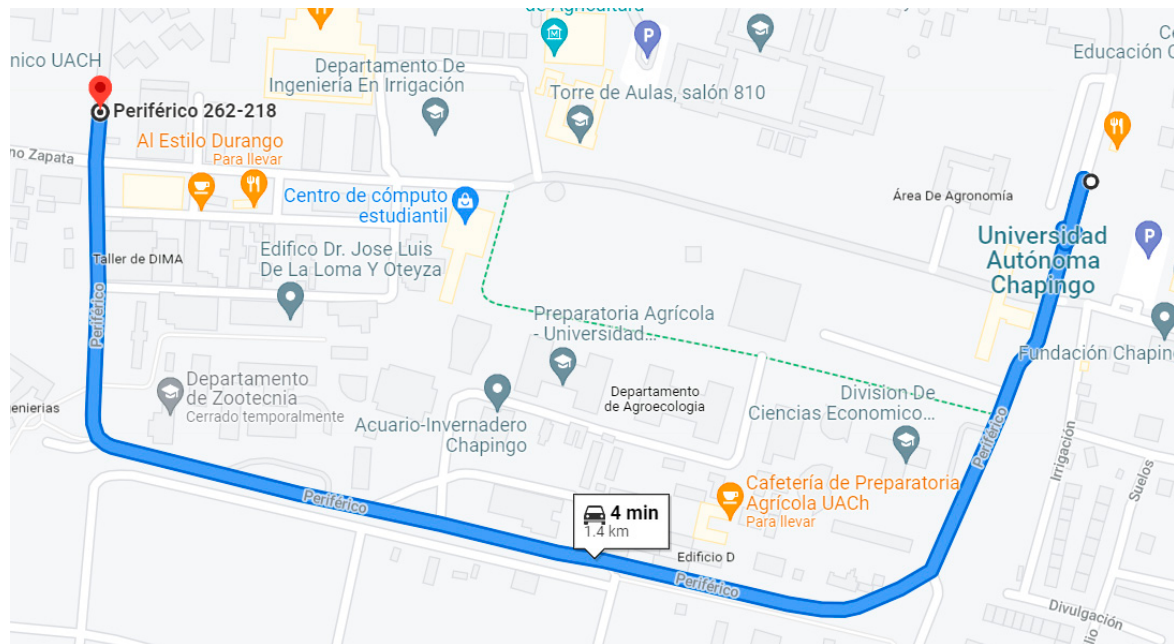


Figure 2. Test scenario with eight WiFi access points with a coverage area of 100 m.

4.1. Network Selection Process according to the Proactive Cross-Layer Protocol

The network selection process is based on the past, present, and future data modeled by the following block diagram of Figure 3.

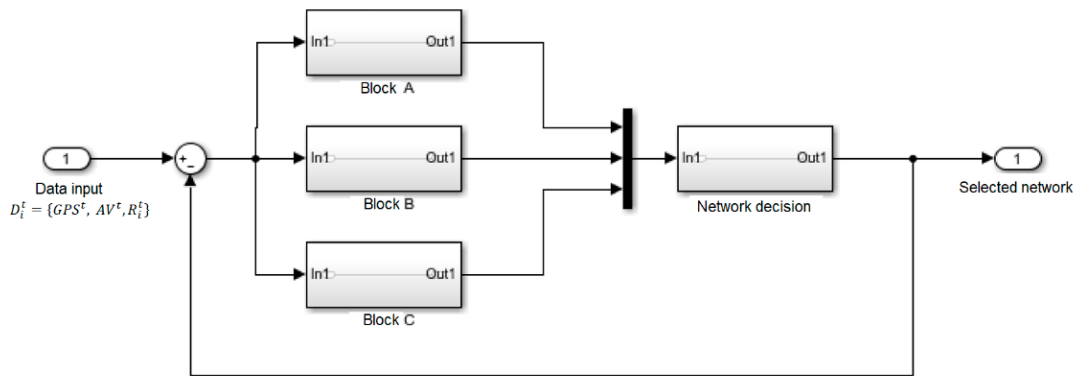


Figure 3. Proactive cross-layer framework block diagram.

In block A, the historical data is used to know where the mobile device has been physically connected. That is, it uses the GPS data set and GPS and R_i^t .

In block B, the current data is sampled in order to analyze the current state of the system and make a compensation if it is necessary. That is, it uses the Kalman filter that was proposed by Rudolf E. Kalman in 1960, where a set of mathematical equations efficiently estimates the state of a linear system that minimizes the estimated error to achieve the optimization. The state of vector x is defined by the set of data $[p_x, v_x, a_x, p_y, v_y, a_y]$, which describes the movements of the mobile in time t where p_x, v_x, a_x corresponds to the latitude point, velocity, and acceleration, and p_y, v_y, a_y corresponds to the parameter length of a mobile. For the measure of the change in the state of the mobile without Δt , we use the kinetic equation of movement (Kalman, 1960). This block is supported by the technique using the AV data set. Next, it uses Equation (1), to draw a possible route and anticipate a possible network change.

$$x = x_0 + v\Delta t + 0.5a^2\Delta t \quad (1)$$

$$v = v_0 + a\Delta t \quad (2)$$

where:

x_0 is the initial state of the vehicle, v is the velocity, and a is the acceleration of the vehicle (assuming constant acceleration at time t).

In block C, a forecast of a possible future location is calculated and, using classification artificial intelligence techniques, it is predicted which the possible network is that will be reached. In this block, a bayesian classifier is used.

4.1.1. Model of Naive Bayes

Naive bayes is a model for classification based on bayes theorem. This model estimates the per-class probability by assuming that the attributes are conditionally independent. Let C denote defined class variable, in this case C_i is WLAN network, E_i denote conditional attribute I used for classification, and $P(C|E_1, \dots, E_n)$ denote conditional probability of class C , given that features of GPS^t and AV^t have happened. The probability model for naive bayes classifier can be defined as shown in Equation (3).

$$P(C|E_1, \dots, E_n) = \frac{P(C)P(E_1, \dots, E_n|C)}{P(E_1, \dots, E_n)} \quad (3)$$

In case of independence, assumption was assumed instead of computing the conditional probability for every combination of attributes E_i given class C . We can derive the Equation (3) as follows:

$$P(C|E_1, \dots, E_n) = \frac{P(C)P(E_1|C), \dots, P(E_n|C)}{P(E_1, \dots, E_n)} \quad (4)$$

where:

C_i are the available WLAN networks to change.

$E_1, \dots, E_n$ are the features of GPS^t and AV^t .

From a classification point of view, Equations (3) and (4) are also called the likelihood functions when they are expressed as a function of C given E . For class variables $C_1, \dots, C_m$ we can classify the evidence into m value of likelihood. Then, we assign the evidence to the class with maximum likelihood. When we compare likelihood function, we can filter out $P(E_1, \dots, E_n)$ from Equation (4) because it is constant. Therefore, let us rewrite the likelihood function as follows:

$$Likelihood(C_i) = P(C_i)P(E_1|C_i), \dots, P(E_n|C_i) \quad (5)$$

Equation (5) is more practical because it does not require a very large training set. The reason is evidence was partitioned into multiple attributes by independence assumption. In other words, it allows the class conditional densities to be calculated separately for each attribute and reduces a multidimensional task to a number of one-dimensional tasks. Therefore, the next network to switch is NR_i .

$$NR_i = \arg\text{Max}(Likelihood(C_i)) \quad (6)$$

where:

C_i are the available WLAN networks to change.

NR_i is the next network to switch to.

It is important to mention that the present data will become historical but the forecast data will not necessarily be part of the historical data set.

4.1.2. Model of Support-Vector Machines

Support-vector machines (SVM) were developed to solve the classification problem and regression problems. As a powerful computational intelligence theory, SVM developed the foundations in 1995 by Vapnik. Regarding the classification problem, two classes are separated by a hyperplane.

$$wx + b = 0 \quad (7)$$

where:

w is the weight matrix.

b is the bias matrix.

x is feature input.

The set of vectors is said to be optimally separated by the hyperplane if it is separated without error and the distance between the closest vectors to the hyper-plane is maximal. Therefore, Equation (8) is as follows.

$$\min \left\{ \frac{1}{2} \|w\|^2 \right\} \quad (8)$$

4.1.3. Model of Multinomial Logistic Regression

A model of classification logistic regression is used for data in which the dependent variable is unordered or polytomous, and independent variables are continuous or categorical predictors. Under a multinomial logistic regression model, the probability that x belongs to class i is written as:

$$(y^{(i)} = 1 | x, w) = \frac{\exp(w^{(i)T} x)}{\sum_{y=1}^m \exp(w^{(y)T} x)} \quad (9)$$

For $i \in \{1, \dots, m\}$, where:

$w^{(i)}$ is the weight vector corresponding to class i , in this case the class is a network and the superscript T denotes matrix transpose. For binary problems ($m = 2$), this is known as a logistic regression model; for $m > 2$, the usual designation is multinomial logistic regression (or soft-max in the neural networks literature) because of the normalization condition.

$$\sum_{y=1}^m P(y^{(i)} = 1 | x, w) = 1 \quad (10)$$

The weight vector for one of the classes need not be estimated. Without loss of generality, we thus set $w^{(m)} = 0$, and the only parameters to be learned are the weight vectors $w^{(i)}$ for $i \in \{1, \dots, m-1\}$.

4.1.4. Model of Decision Tree

The decision tree algorithm works very well and continuously if the data are discontinuous, even if noise appears. Moreover, it can handle collinearity efficiently and provide excellent prediction explanation. On the other hand, DTs suffer from higher complexity, especially when dealing with complicated datasets, and consequently they may lose valuable information in case of continuous variables.

The decision tree model aims to find the best split node that guarantees high accuracy. The information gain (IG) method seeks to find the most suitable nodes that return the highest information gain, which can be measured using an entropy factor. The entropy factor is used to determine the degree of disorganization in the system. The entropy for the output can be calculated using the following formula:

$$E(s) = \sum_{i=1}^c -p_i \log_2(p_i) \quad (11)$$

where p_i : proportion of samples that belongs to class c for a particular node, and C is the class, in this case the available WLAN networks to change.

4.1.5. Model of k-Nearest Neighbors

The k-nearest neighbors (KNN) model is one of the simplest and most straightforward supervised learning models in machine learning. The key idea of this algorithm is to decide a predicted value based on the labeled data points of the training set that are near the query data point.

KNN starts by loading the training data points in memory. Then, the classification task is completed by finding the nearest k data points. Finally, a vote of the k closest points to the query point will determine the class of the query data point. One critical decision that needs to be made is the selection of the distance function. Several distance functions have been proposed to compute the distance between two data points; however, the most common methods are cosine similarity and Euclidean distance.

The Euclidean distance can be calculated by subtracting the training data point from the point to be classified, as in Equation (12).

$$E(x, y) = \sqrt{\sum_{i=0}^n (x_i - y_i)^2} \quad (12)$$

In this calculation, we determine the data point class based on the best posterior probability value. Although KNN contains a limited number of hyper-parameters (i.e., the k -value and distance function), which makes it a simple model, the K -value can dramatically affect the model performance.

Five classification models were compared to each other to get the best fit for the data set on the network. They are the support-vector machines (SVM), naive bayes (NB), logistic regression (LR), decision tree (DT), and k-nearest neighbors (KNN).

4.2. System Architecture

The system uses native APIs operating system android, WiFi, location, and sensors, such as native supplicants, to obtain the data and store it in a separated file, as illustrated in Figure 4.

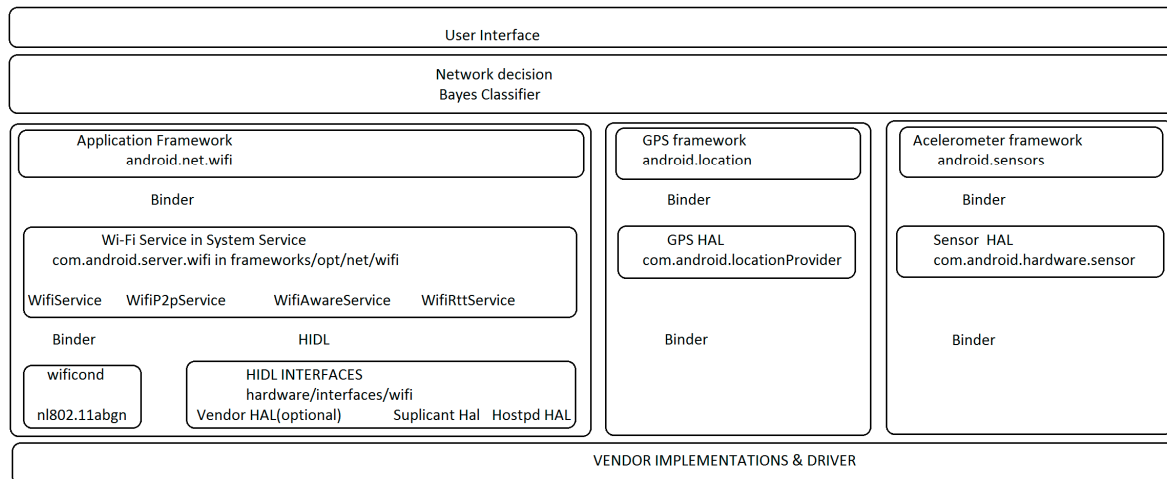


Figure 4. System architecture.

4.3. Decision Taking Procedure

The system takes a sample of the position, acceleration, and available networks every second. Then, based on the data set, it predicts which the next network is, as illustrated in Figure 5.

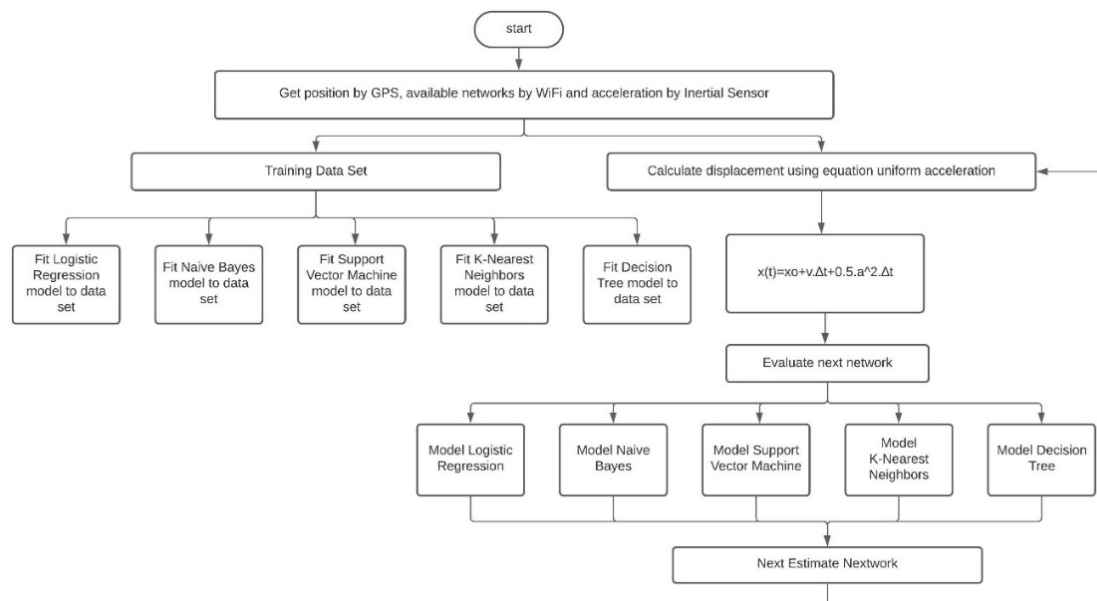


Figure 5. System flow diagram.

To collect the data, an application for mobile devices is used. Then the data is exported to a CSV file and processed with a MATLAB script. Finally, the next network is forecast.

4.4. Data Acquisition

The data was collected using an application for android mobile devices programmed in android studio 3.5.2. Basically, the application scans the network environment, the GPS position, speed, and acceleration every 5 s and stores it in a CSV file extension. The user

interface of the application is very simple because it only shows the results every second, as illustrated in Figure 6.

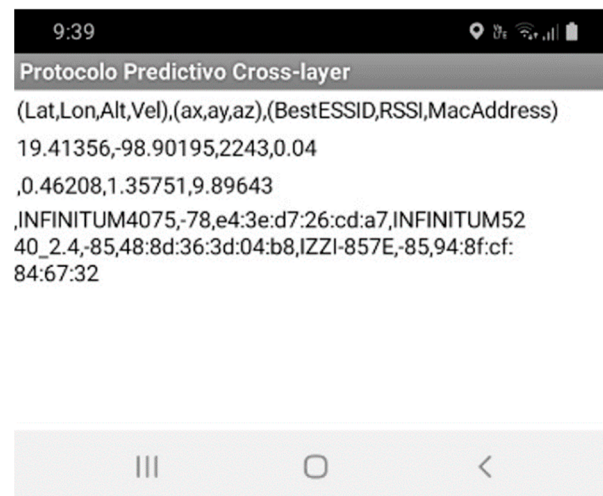


Figure 6. Screenshot smartphone.

Next, the data is stored in an archive csv root file in the internal storage of the mobile device. The organization of the resulting file is by columns: latitude, longitude, height, velocity, acceleration x axis, y axis and z axis, best ESSID, RSSI, mac address, and other available networks following the same sequence.

This application allows viewing the proximity of the access point in four dimensions. The longitude, latitude, and height dimensions are represented on the x , y , and z axis, respectively. The fourth dimension is the power of the received signal and is represented by a colored bar. Below are different views of the graph obtained for a single access point. Please see Figure 7.

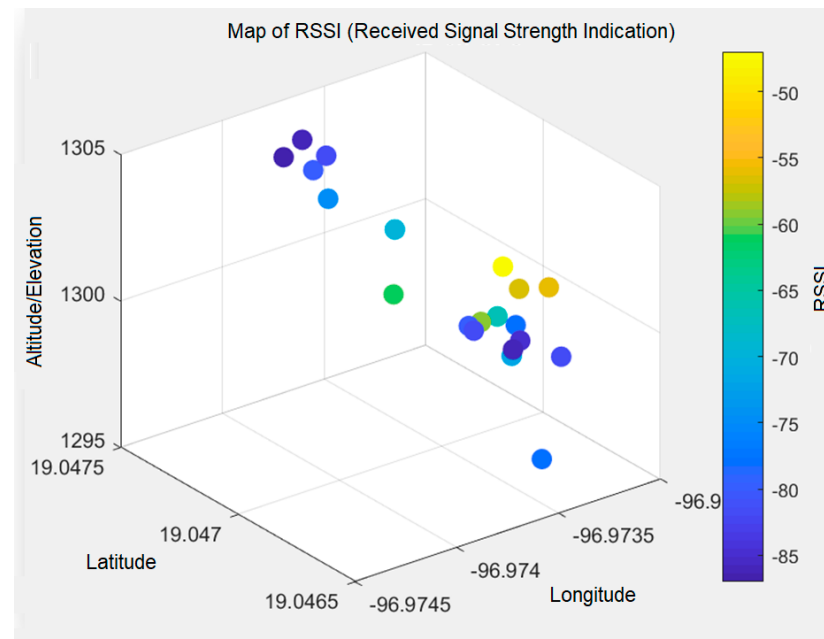


Figure 7. 4D Graph of the access point with position and received signal strength.

Note that closer to the access point the power increases, and further from it, the power decreases. This application for mobile devices can also be used to map the power of the received signal.

4.5. Implementation and Hyper-Parameters

The software is programmed in python3.8 using the sklearn API and IDE Spyder by Anaconda, running on omen HP laptop, whit Intel Core i7 7th gen, 16 GB of memory RAM, and NVIDIA GeForce GTX 1050 4GB. The KNN classifier uses 8 neighbors, for the naive bayes classifier using Gaussian naive bayes, for the decision tree, for the support-vector machines, and for the logistic regression classifiers using default parameters. The hyper-parameters set to KNN is Euclidean distance and $k = 8$, the other classifiers set default hyper-parameters without any hyper-parameter tuning. Table 2 shows the result from instruction “get_params”, which returns all hyper-parameters from each model.

Table 2. Model of hyper-parameters.

Technique	Hyper-Parameters from Sklearn
LR	{‘C’: 1.0, ‘class_weight’: None, ‘dual’: False, ‘fit_intercept’: True, ‘intercept_scaling’: 1, ‘l1_ratio’: None, ‘max_iter’: 100, ‘multi_class’: ‘auto’, ‘n_jobs’: None, ‘penalty’: ‘l2’, ‘random_state’: None, ‘solver’: ‘lbfgs’, ‘tol’: 0.0001, ‘verbose’: 0, ‘warm_start’: False}
NB	{‘priors’: None, ‘var_smoothing’: 1e-09}
SVM	{‘C’: 1.0, ‘break_ties’: False, ‘cache_size’: 200, ‘class_weight’: None, ‘coef0’: 0.0, ‘decision_function_shape’: ‘ovr’, ‘degree’: 3, ‘gamma’: ‘scale’, ‘kernel’: ‘rbf’, ‘max_iter’: -1, ‘probability’: False, ‘random_state’: None, ‘shrinking’: True, ‘tol’: 0.001, ‘verbose’: False}
KNN	{‘algorithm’: ‘auto’, ‘leaf_size’: 30, ‘metric’: ‘minkowski’, ‘metric_params’: None, ‘n_jobs’: None, ‘n_neighbors’: 8, ‘p’: 2, ‘weights’: ‘uniform’}
DT	{‘ccp_alpha’: 0.0, ‘class_weight’: None, ‘criterion’: ‘gini’, ‘max_depth’: None, ‘max_features’: None, ‘max_leaf_nodes’: None, ‘min_impurity_decrease’: 0.0, ‘min_impurity_split’: None, ‘min_samples_leaf’: 1, ‘min_samples_split’: 2, ‘min_weight_fraction_leaf’: 0.0, ‘presort’: ‘deprecated’, ‘random_state’: None, ‘splitter’: ‘best’}

5. Results

In the scenario of Figure 8, one hundred samples were taken per access point and then 50% were taken for training and the rest for testing to use cross validation. Figure 8 graphs the data that was used for training, and the data that was used for testing. In addition, the data that was correctly classified is very close to the access point to which it belonged. Therefore, the accuracy must be very high. In the following, graph samples are taken very far from the access point and the accuracy of the classifier decreases drastically, as shown in Figure 8.

This is because there is a balance between the train data and the test data. Figure 9 shows the data obtained with a graph between the accuracy of the algorithm, and Table 3 shows the percentage of data used to test the different algorithms.

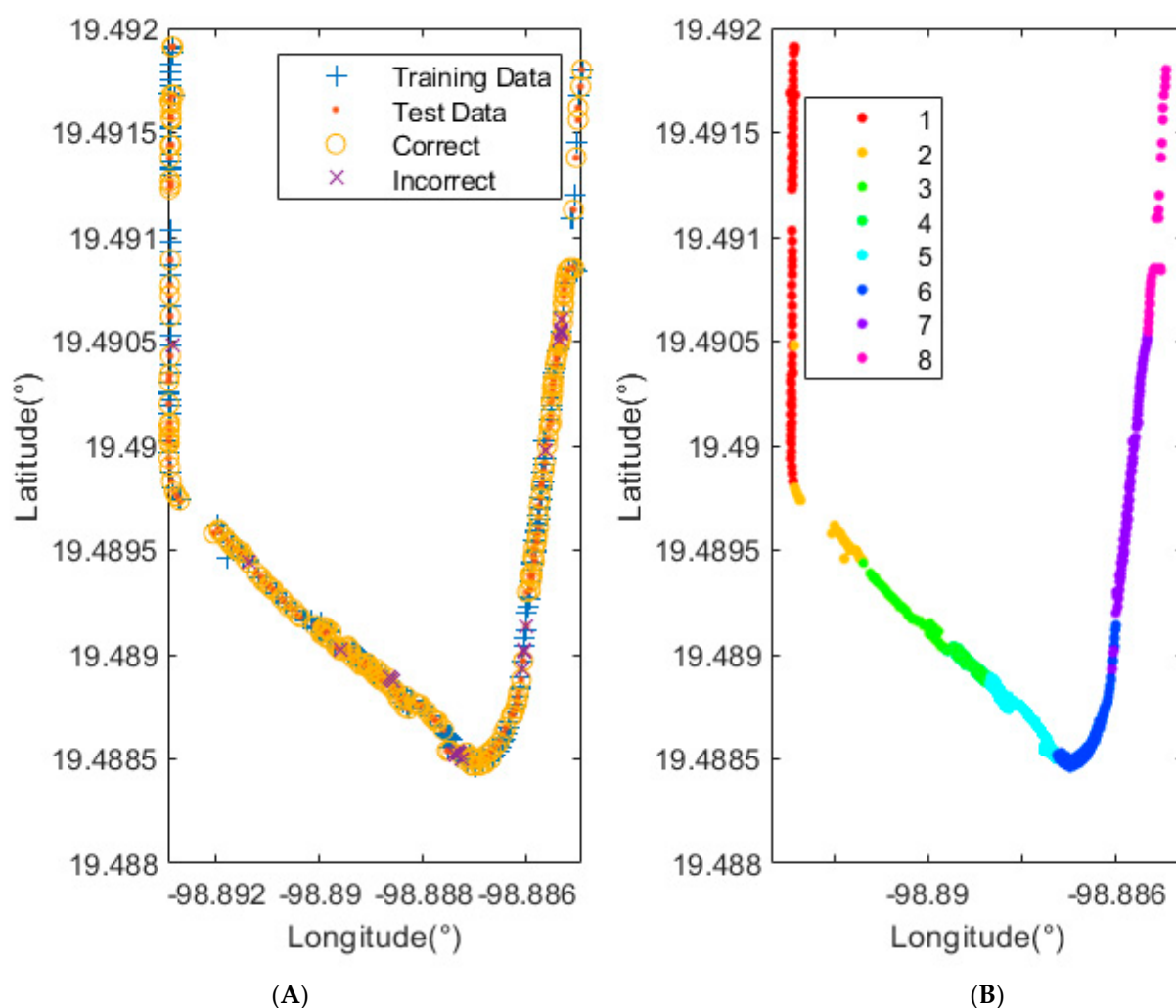


Figure 8. (A). Split date test and train each independent data, accuracy = 88.92%. (B). Graph all data and class.

Table 3. Comparison of techniques against the percentage of data testing.

Technique	50%	40%	30%	20%
LR	34.74%	36.99%	35.68%	36.59%
NB	82.14%	84.15%	82.16%	86.99%
SVM	21.43%	16.67%	12.43%	16.26%
KNN	67.53%	68.70%	70.81%	75.61%
DT	95.13%	96.34%	95.14%	96.75%

Figure 9 and Table 3 show the relationship between the test data and accuracy. We observe from the previous graph that the algorithm that best fits the data is the decision tree, followed by naive bayes, k-nearest neighbors, logistic regression, and finally the support-vector machines.

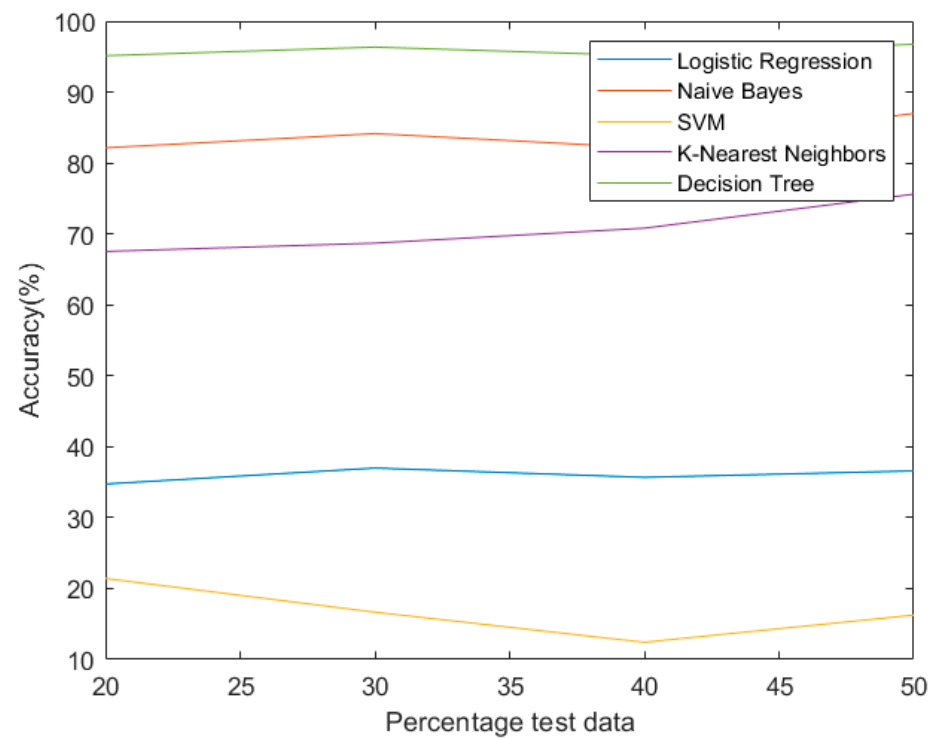


Figure 9. Percentage accuracy vs. percentage test data.

In Figure 10, the graph grouped by classifier is shown and the results obtained from the least amount of data to the greatest amount of test data are also shown from right to left—purple = 20%, orange = 30%, red = 40%, and blue = 50% data test. As can be seen, the greater the amount of test data, the accuracy decreases, and in the center where the data is equally proportional, the accuracy is similar.

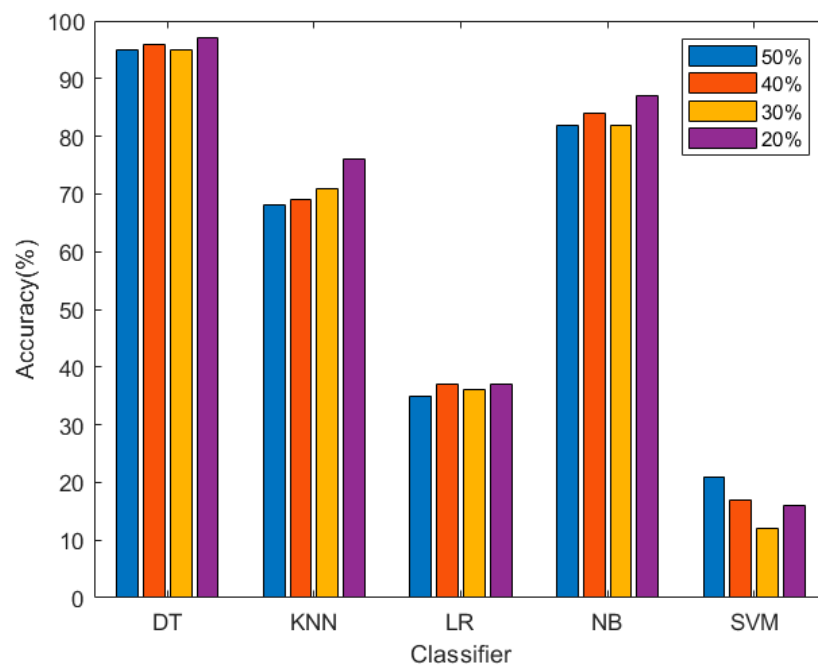


Figure 10. Accuracy distribution group by technique.

6. Discussion

This section describes the discussion of handover decision, which is divided into two parts: (A) The mobility forecast part, in which usually the articles consulted use techniques for estimating the position of a SL, and (B) The part of classification techniques to determine which characteristics belong to a class.

With regard to mobility prognosis, the techniques used in this thesis are Kalman filter (KF), particle filter (PF), and artificial neural networks (ANN).

The specialized literature usually uses different filters to estimate the position of an NM as in (Yan, 2019), which proposes a CSI scheme (Canal State Information) contrasting its results with a Kalman filter of [33], please see Figure 11. This proposal obtains an error improvement in the x axis of 0.63 m and in the y axis of 0.75 m, considering that the node moves in a margin in the x axis of 100 m and on the y -axis of 8 m. An error of 0.63% on the x axis and 9.37% on the y axis is estimated.

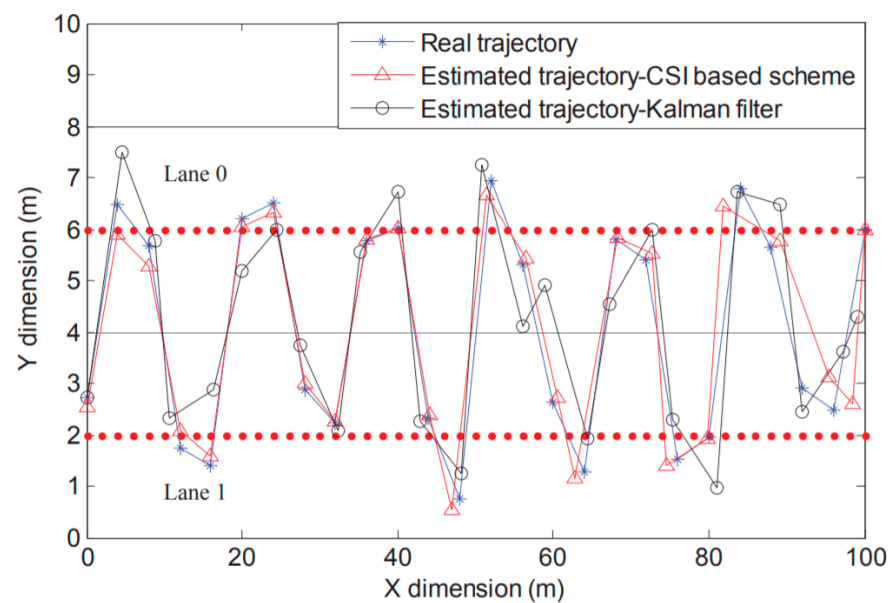


Figure 11. Estimated trajectory graph comparison.

In contrast to the proposal presented in this paper, which was tested with a movement of 2000 m on the x -axis and 40 m on the y -axis, it yielded an average error of 33.58% in the artificial neural network and 45.74% in the filter of particles. Considering that it is a random movement and that in theory it does not have any movement pattern, it was decided to define a test scenario as in all the articles consulted in this research. In this thesis, a test scenario with uniformly accelerated movement was defined in a circuit with seven access points.

Regarding the classification techniques used, five classifiers with seven input characteristics and one output with seven possible networks are proposed. Figure 12 shows a set of graphs that are contrasted with each other, that is, in the first row latitude is plotted on the y -axis against the remaining characteristics in order to observe their distribution and to know if they are linearly separable. On the other hand, if acceleration versus velocity is plotted, it can be seen that the points are very close; therefore, a classifier with these characteristics would not perform well.

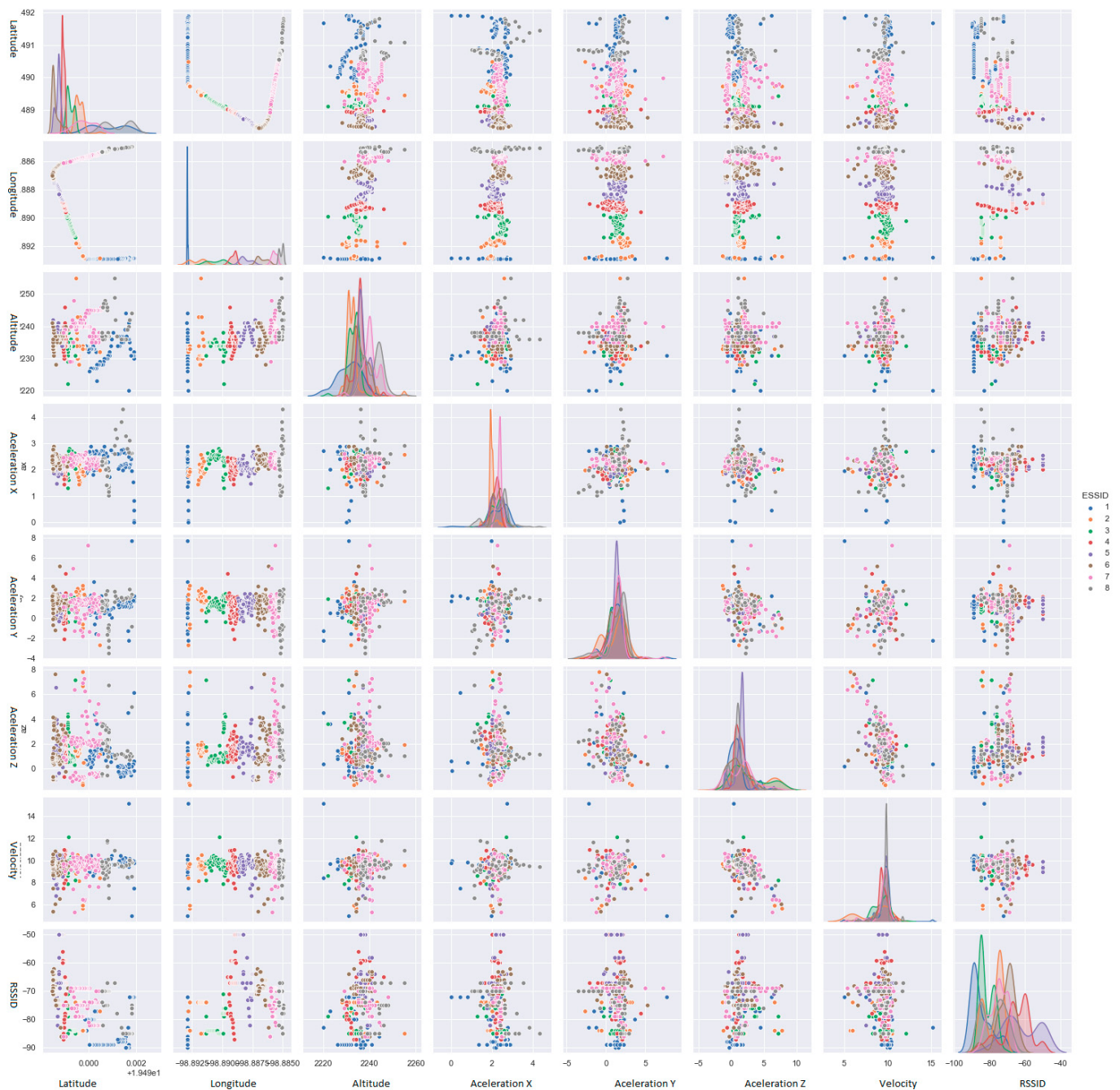


Figure 12. Estimated feature comparison chart.

Figure 12 diagonally shows the distribution of the eight variables of our data set. In other cells of the plot matrix, we have correlation plots of each variable combination of our data frame. In the first row and second row we can see latitude on ordinate and longitude on abscissa, and the right graph shows the correlation between latitude on ordinate and altitude on abscissa and then each combination to other variables.

Support-vector machines work very well with a lot of data because they can have more support-vectors; however, in two case studies the classifier with similar behavior in percentage accuracy is the naive bayes classifier with an approximate accuracy of 80%. For this reason we recommend the naive bayes classifier as the best option because the other classifiers vary depending on the amount of data, data distribution, or imbalanced data.

Figure 13 shows the confusion matrix results.

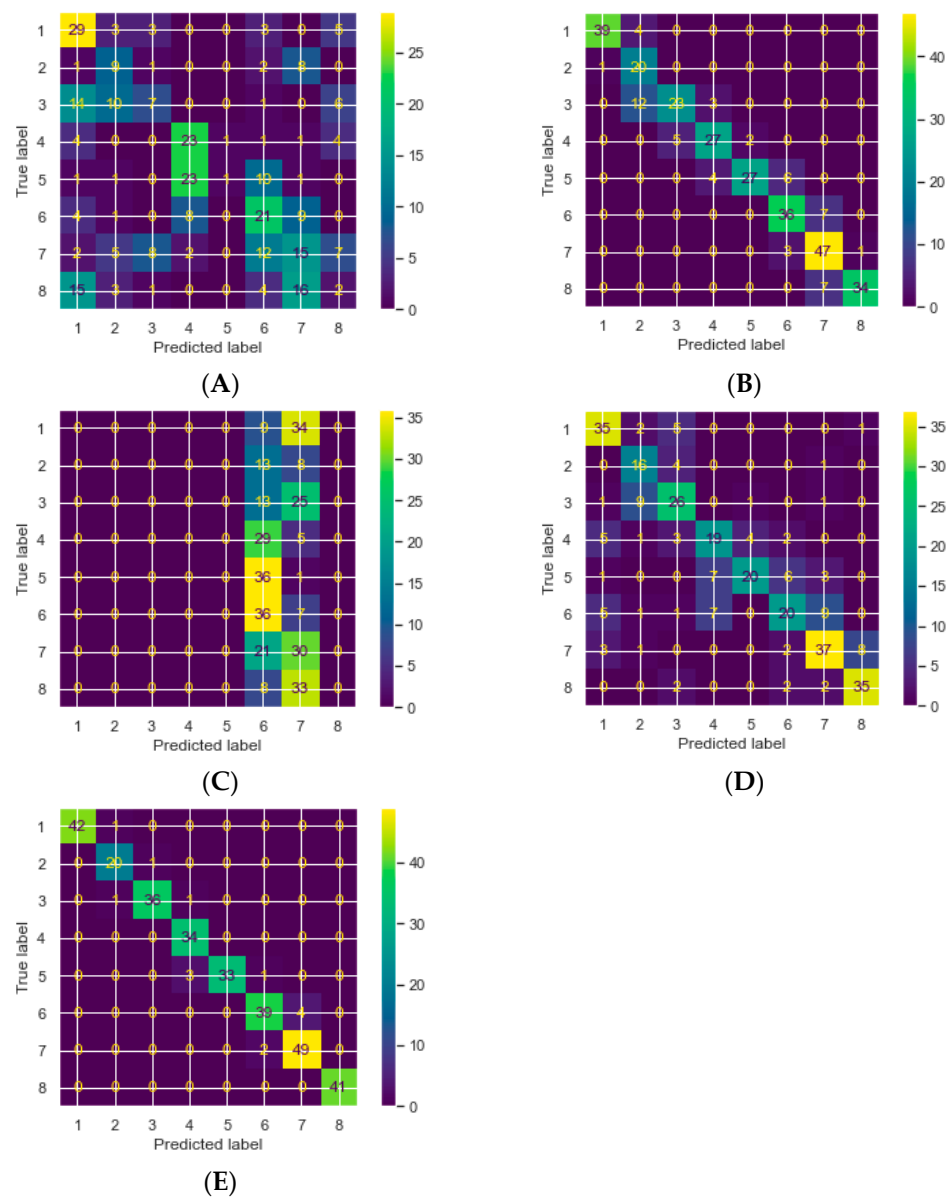


Figure 13. Confusion matrix of the model's comparison chart. (A) Confusion matrix of logistic regression. (B) Confusion matrix of naive bayes. (C) Confusion matrix of SVM. (D) Confusion matrix of KNN. (E) Confusion matrix of DT.

The confusion matrices (B), (D), and (E) on Figure 13 have similar behaviors, which indicates that the NB, KNN, and DT models fit better than LR and SVM to the data set.

The time to train and test models is presented in Table 4. NB, KNN, and DT have similar behavior, while SVM and LR take longer time to fit the model to the data set.

Table 4. Comparison of training and testing time.

Technique	Training Time (ms)	Testing Time (ms)
LR	33.9	1.0
NB	1.0	1.0
SVM	6.0	3.0
KNN	1.0	9.0
DT	1.0	1.0

Other proposals in the literature have been developed with similar results as shown in Table 5, in similar scenarios, but this approach is part of an application for mobile devices that will be added in the future with geo-fences to delimit the behavior of the mobile node.

Table 5. Comparison of results obtained versus handover decision algorithms.

Approach	Title	Year	Clasificador	Network	Features Input	Accuracy
(Wang, 2014)	SVM-based spectrum mobility Prediction scheme in mobile cognitive radio networks	2014	SVM	Mobile cognitive radio networks	Velocity, position	95%
(Ma, 2017)	Modeling and analysis for vertical handoff based on the decision tree in a heterogeneous vehicle network	2017	Decision tree (DT)	heterogeneous Network	Received signal strength (RSS), Bit error rate (VER), Brocking rate (BR)	95%
(Svahn, 2019)	Inter-frequency radio signal quality prediction for handover, evaluated in 3GPP LTE.	2019	Decision tree (DT)	LTE 3GPP	Reference signal received quality (RSRQ)	90%
(Ryder, 2018)	Predicting strongest cell on secondary carrier using primary carrier data	2018	Decision tree (DT)	LTE	Reference signal received quality (RSRQ)	95%
(Our Approach)	Proactive cross-layer framework based on classification techniques for handover decision on WLAN environments	2022	KNN, SVM, DT, NB y RL	WLAN	position, acceleration, velocity, Received signal strength (RSS)	96%

7. Conclusions and Future Work

This section describes the conclusions that have been obtained from this research. With respect to the techniques estimating the next access point (AP) being 92% average accuracy, while the next AP is correct, this implies a seamless handover without data lost. An exponential increase in the acquisition time of an IP address was also observed depending on the signal strength. Therefore, performing a jump with a power greater than -40 db or 60% of quality of link is recommended. As future work, the implementation of this algorithm in a streaming data, or voice data call, to measure the quality of experience will be applied.

Author Contributions: Investigation and formal analysis J.V.C.-B., L.A.S. and L.M.R.-C.; software and validation S.L.-S., F.G.-L. and A.D.C.-R.; writing—original draft, review & editing J.d.J.R. and J.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: Authors appreciate the infrastructure provided by Autonomous University of the State of Mexico (UAEMex), Metropolitan Autonomous University (UAM) and Autonomous University Chapingo (UACH), because this research could not be done without your valuable support. Authorsthankthe Instituto Politécnico Nacional, Secretaría de Investigación y Posgrado, Comisión de Operación y Fomento de Actividades Académicas, and Consejo Nacional de Ciencia y Tecnología for their help in this research.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Wang, X.; Li, X.; Leung, V.C. Artificial intelligence-based techniques for emerging heterogeneous network: State of the arts, opportunities, and challenges. *IEEE Access* **2015**, *3*, 1379–1391. [\[CrossRef\]](#)
- Khan, M.; Din, S.; Gohar, M.; Ahmad, A.; Cuomo, S.; Piccialli, F.; Jeon, G. Enabling multimedia aware vertical handover Management in Internet of Things based heterogeneous wireless networks. *Multimed. Tools Appl.* **2017**, *76*, 25919–25941. [\[CrossRef\]](#)
- Kassar, M.; Kervella, B.; Pujolle, G. An overview of vertical handover decision strategies in heterogeneous wireless networks. *Comput. Commun.* **2008**, *31*, 2607–2620. [\[CrossRef\]](#)
- Nair, M.; Zhu, F. Vertical handoffs in fourth-generation multinet environments. *IEEE Wirel. Commun.* **2004**, *11*, 8–15.
- Ghanem, K.; Alradwan, H.; Motermawy, A.; Ahmad, A. Reducing ping-pong Handover effects in intra EUTRA networks. In Proceedings of the 2012 8th International Symposium on Communication Systems, Networks & Digital Signal Processing, Poznan, Poland, 18–20 July 2012; pp. 1–5.
- Tsai, K.L.; Liu, H.Y.; Liu, Y.W. Using fuzzy logic to reduce ping-pong handover effects in LTE networks. *Soft Comput.* **2016**, *20*, 1683–1694. [\[CrossRef\]](#)
- Majlesi, A.; Khalaj, B.H. An adaptive fuzzy logic based handoff algorithm for interworking between WLANs and mobile networks. In Proceedings of the The 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Lisbon, Portugal, 18 September 2002; Volume 5, p. 2.
- Xia, L.; Jiang, L.G.; He, C. A novel fuzzy logic vertical handoff algorithm with aid of differential prediction and pre-decision method. In Proceedings of the IEEE International Conference on Communications, Glasgow, UK, 24–28 June 2007; pp. 5665–5670.
- Kustiawan, I.; Liu, C.Y.; Hsu, D.F. Vertical Handoff Decision Using Fuzzification and Combinatorial Fusion. *IEEE Commun. Lett.* **2017**, *21*, 2089–2092. [\[CrossRef\]](#)
- Wang, Y.; Zhang, Z.; Ma, L.; Chen, J. SVM-based spectrum mobility prediction scheme in mobile cognitive radio networks. *Sci. World J.* **2014**, *2014*, 395212. [\[CrossRef\]](#)
- Ma, B.; Wang, D.; Cheng, S.; Xie, X. Modeling and analysis for vertical handoff based on the decision tree in a heterogeneous vehicle network. *IEEE Access* **2017**, *5*, 8812–8824. [\[CrossRef\]](#)
- Svahn, C.; Sysoev, O.; Cirkic, M.; Gunnarsson, F.; Berglund, J. Inter-frequency radio signal quality prediction for handover, evaluated in 3GPP LTE. In Proceedings of the 2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring), Kuala Lumpur, Malaysia, 28 April–1 May 2019; pp. 1–5.
- Ryden, H.; Berglund, J.; Isaksson, M.; Cöster, R.; Gunnarsson, F. Predicting strongest cell on secondary carrier using primary carrier data. In Proceedings of the 2018 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), Barcelona, Spain, 15–18 April 2018; pp. 137–142.
- Zheng, Q.; Yang, M.; Yang, J.; Zhang, Q.; Zhang, X. Improvement of generalization ability of deep CNN via implicit regularization in two-stage training process. *IEEE Access* **2018**, *6*, 15844–15869. [\[CrossRef\]](#)
- Jin, B.; Cruz, L.; Gonçalves, N. Face depth prediction by the scene depth. In Proceedings of the 2021 IEEE/ACIS 19th International Conference on Computer and Information Science (ICIS), Shanghai, China, 23–25 June 2021; pp. 42–48.
- Jin, B.; Cruz, L.; Gonçalves, N. Deep facial diagnosis: Deep transfer learning from face recognition to facial diagnosis. *IEEE Access* **2020**, *8*, 123649–123661. [\[CrossRef\]](#)
- Tanveer, J.; Haider, A.; Ali, R.; Kim, A. An Overview of Reinforcement Learning Algorithms for Handover Management in 5G Ultra-Dense Small Cell Networks. *Appl. Sci.* **2022**, *12*, 426. [\[CrossRef\]](#)
- Zheng, Q.; Yang, M.; Tian, X.; Jiang, N.; Wang, D. A full stage data augmentation method in deep convolutional neural network for natural image classification. *Discret. Dyn. Nat. Soc.* **2020**, *2020*, 4706576. [\[CrossRef\]](#)
- Liao, Y.; Cao, L. Practical schemes for smooth MAC layer handoff in 802.11 wireless networks. In Proceedings of the 2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'06), Buffalo-Niagara Falls, NY, USA, 26–29 June 2006; pp. 1–10.
- Mishra, A.; Shin, M.; Arbaugh, W. An empirical analysis of the IEEE 802.11 MAC layer handoff process. *ACM SIGCOMM Comput. Commun. Rev.* **2003**, *33*, 93–102. [\[CrossRef\]](#)
- Li, C.S.; Tseng, Y.C.; Chao, H.C.; Huang, Y.M. A neighbor caching mechanism for handoff in IEEE 802.11 wireless networks. *J. Supercomputing.* **2008**, *45*, 1–14. [\[CrossRef\]](#)
- Santi, S.; De Koninck, T.; Daneels, G.; Lemic, F.; Famaey, J. Location-Based Vertical Handovers in Wi-Fi Networks With IEEE 802.11 ah. *IEEE Access* **2021**, *9*, 54389–54400. [\[CrossRef\]](#)
- Chen, Y.; Niu, K.; Wang, Z. Adaptive handover algorithm for LTE-R system in high-speed railway scenario. *IEEE Access* **2021**, *9*, 59540–59547. [\[CrossRef\]](#)
- Kohl, N.; Mikkilainen, R. Evolving neural networks for strategic decision-making problems. *Neural Netw.* **2009**, *22*, 326–337. [\[CrossRef\]](#) [\[PubMed\]](#)
- Nasser, N.; Guizani, S.; Al-Masri, E. Middleware vertical handoff manager: A neural network-based solution. In Proceedings of the IEEE International Conference on Communications, Glasgow, UK, 24–28 June 2007; pp. 5671–5676.
- Mahira, A.G.; Subhedar, M.S. Handover Decision in Wireless Heterogeneous Networks Based on Feedforward Artificial Neural Network. *Comput. Intell. Data Mining.* **2017**, *556*, 663–669.

27. Chen, W.; Gong, S.; Jiang, X. Fuzzy multiple attribute decision access scheme in heterogeneous wireless network. *Multimed. Tools Appl.* **2017**, *76*, 20049–20065. [[CrossRef](#)]
28. Fei, L.; Zhang, B.; Xu, Y.; Tian, C.; Rida, I.; Zhang, D. Jointly Heterogeneous Palmprint Discriminant Feature Learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, 1–12. [[CrossRef](#)]
29. Fei, L.; Zhang, B.; Tian, C.; Teng, S.; Wen, J. Jointly learning multi-instance hand-based biometric descriptor. *Inf. Sci.* **2021**, *562*, 1–12. [[CrossRef](#)]
30. Wen, J.; Sun, H.; Fei, L.; Li, J.; Zhang, Z.; Zhang, B. Consensus guided incomplete multi-view spectral clustering. *Neural Netw.* **2021**, *133*, 207–219. [[CrossRef](#)] [[PubMed](#)]
31. Marquez, G.; Taramasco, C.; Astudillo, H.; Zalc, V.; Istrate, D. Involving Stakeholders in the Implementation of Microservice-Based Systems: A Case Study in an Ambient-Assisted Living System. *IEEE Access* **2021**, *9*, 9411–9428. [[CrossRef](#)]
32. Tylcz, J.-B.; Muszynski, C.; Dauchet, J.; Istrate, D.; Marque, C. An Automatic Method for the Segmentation and Classification of Imminent Labor Contraction From Electrohysterograms. *IEEE Trans. Biomed. Eng.* **2020**, *67*, 1133–1141. [[CrossRef](#)] [[PubMed](#)]
33. Yan, L.; Ding, H.; Zhang, L.; Liu, J.; Fang, X.; Fang, Y.; Xiao, M.; Huang, X. Machine Learning-Based Handovers for Sub-6 GHz and mm Wave Integrated Vehicular Networks. *IEEE Trans. Wirel. Commun.* **2019**, *18*, 4873–4885. [[CrossRef](#)]