Comparing different Machine-Learning techniques to predict Vehicles' Positions using the received Signal Strength of periodic messages

Mamoudou Sangare, Dinh-Van Ngdvan Nguyen, Soumya Banerjee, Paul Mühlethaler, Samia Bouzefrane

Comparing different Machine-Learning techniques to predict Vehicles' Positions using the received Signal Strength of periodic messages

Mamoudou Sangare*, Dinh Van Nguyen† Soumya Banerjee‡, Paul Muhlethaler*, Samia Bouzefrane§
*INRIA EVA, Centre de Recherche de Paris, 2 Rue Simone, IFF CS 42112, 75589 Paris Cedex 12
Email: {paul.muhlethaler}@inria.fr
*INRIA RITS, Centre de Recherche de Paris, 2 Rue Simone, IFF CS 42112, 75589 Paris Cedex 12
Email: {dinh-van.nguyen}@inria.fr
‡ Department of Computer Science and Engeneering. Birla Institute of Technology, Mesra, India
Email: {dr.soumya}@ieee.org
§ CEDRIC Lab, CNAM, 292 Rue Saint-Martin, 75003 Paris, France
Email: samia.bouzefrane@lecnam.net

***Abstract*—In this paper, vehicles use the beacons sent by Road Side Units (RSUs) to predict their positions on a road. The reception power is strongly influenced by the distance between a vehicle and the neighboring RSUs and thus Machine-Learning can be used to predict the position of vehicles between RSUs. We have to assume that the vehicles know their own positions, at least for a given duration, to build the model of the machine-learning algorithm. This position information can be obtained for instance from a GPS. When this information is no longer available, the machine-learning algorithm can be used to predict the vehicles' positions. The vehicles can send a position request to the RSUs which will know the reception power of their beacons and the machine-learning algorithm can respond with the estimated position. In this study, we compare four well-known machine-learning techniques : K Nearest Neighbors (KNN), Neural Network (NN), Random Forest (RF) and Support Vector Machine (SVM). We study these techniques with different assumptions and discuss their respective advantages and drawbacks. Our results show that these four techniques provide very good results in terms of position predictions when the error on the transmission power is small.**

I. Introduction

In Vehicular Ad Hoc NETworks (VANETs), the vehicles and the RoadSide Units use the IEEE 1609 WAVE (Wireless Access in Vehicular Environments [1]) protocol built on the IEEE802.11p access protocol to provide communication between vehicles (V2V), and between vehicles and roadside infrastructure (V2I).

VANETs are primarily designed to carry communication concerning safety applications. These applications use periodic packet transmissions which carry the speed and the position of the sending vehicles. In Europe we have two types of messages for safety: Car Awareness Messages (CAMs) [2] and Decentralized Environmental Notification Messages (DENMs) [3]. DENMs are multihop broadcasted when a hazardous event occurs on the road, whereas the CAMs sent only at one-hop, carry information about the vehicles' velocities and positions. VANETs can also be used for other purposes. For instance, information about the status of the vehicular traffic such as fluid traffic, traffic jams, etc. can be sent to vehicles. Other less important information can be sent to vehicles or their passengers such as advertising or entertainment, which is usually called infotainment.

VANETs can also be used as a positioning system. In the previous section we have seen that for safety reasons the vehicles send periodic CAM messages which carry their positions and speeds. This information is usually obtained with the GPS; thus if RoadSide Units exist along the road where the vehicles are moving, a huge quantity of positioning data can be made available to machine-learning algorithms. We assume that the vehicles continue to send their CAMs carrying no (or very little) position information. These messages can be used to establish the vehicle's position by using the power at which the RSUs or the vehicles receive the beacons. Machine-learning can be used to perform this task. In this study we will consider four well-known machine-learning techniques namely: K Nearest Neighbors (KNN), Random Forest (RF), Neural Networks and Support Vector Machine (SVM).

In this paper, the contributions are as follows:

- We use the reception power to predict a vehicle's position.
- We propose and adapt four learning techniques to the positioning of vehicles : K Nearest Neighbors (KNN), Neural Network (NN), Random Forest (RF) and Support Vector Machine (SVM).
- A simple simulation tool is developed to produce data with the positions of vehicles and the different powers of messages sent by vehicles and received at the base stations.
- We analyze and compare the performance of K Nearest Neighbors (KNN), Neural Network (NN) Random Forest (RF) and Support Vector Machine (SVM) with the given dataset.

The remainder of this paper is organized as follows: Section II

presents related work. Section III describes the four machine-learning techniques: K Nearest Neighbor (KNN), Neural Network (NN), Random Forest (RF) and Support Vector Machine (SVM) In Section IV, the simulation scenario and numerical results are presented. These different machine-learning techniques and their optimizations are discussed. The methods and their performances are compared. Finally, Section V concludes the paper.

II. Related work

Three main techniques are used for outdoor positioning which are given below:

- Time-Of-Arrival (TOA)-based techniques. These techniques rely on the measurements of the distance between the receiver of a signal and the base station. The position is computed using a triangulation technique. Precised time measurements are not possible for VANETs, mostly because these techniques require a perfect synchronization between the clocks of the base stations and the receivers. To obtain this result, the use of atomic clocks which are very expensive is required. TOA is however the basis of GPS and other similar positioning systems.
- Techniques based on the round trip delay [4]. A small packet is sent by the transmitter to the base station and the sender waits for a reply. The distance between the base station and the sender is proportional to the time elapsed The transmitter can compute its position by triangulation if it can compute three round trip delays from three different base stations. A precise estimation of the location requires a large distance between the sender and the base station. However, like TOA schemes, techniques based on round trip delay require very accurate delay evaluation, which is very difficult unless dedicated transmission modems are used. VANETs use off-the-shelf IEEE 802.11p communication units thus techniques based on the round trip delay are generally not suitable for these networks.
- Signal-strength-based techniques. Based on the received signal strength from several wireless access points the vehicles can compute an estimation of their location. In this case, the road must be completely covered by the access points. Reported results based on this technique show poor accuracy [5], [6].

In vehicular networks the GPS or other similar positioning systems are the most widely used positioning techniques [4] even though they have three main drawbacks: limited accuracy, incomplete coverage and security problems. The accuracy of civilian GPS is around 20 meters, which is not suitable for many VANET applications e.g. lane tracking, collision avoidance, autonomous driving, etc. The best accuracy claimed by GPS vendors is plus/minus 5 meters but this accuracy is achieved for only 95% of the time, leaving the remaining 5% with much larger margins. Thus GPS alone is not suitable for critical applications. GPS coverage is incomplete; GPS has a high accuracy only when four signals can be detected from four different satellites and this situation is quite unlikely even if we do not consider the obvious case of obstruction by, for example, tunnels. Moreover attackers can use strong fake GPS signals that the vehicles are forced to lock on to, which can lead to large errors in the vehicles' positions.

Several contributions have promoted an enhancement of GPS called Differential GPS (DGPS) where transmitters whose locations are precisely known complement the signals sent by the satellites. However, DGPS and other similar techniques do not work when the signals are too weak, for instance in underground, in tunnels, or in densely built-up areas.

III. The machine-learning schemes

We use four widely accepted machine-learning techniques: K Nearest Neighbors (KNN), Neural Networks (NN), Random Forest (RF) and Support Vector Machine (SVM). These four techniques and their suitability to perform positioning in VANETs are recalled below.

In machine-learning schemes, we often have a vector $X_j = \{x_j^1, \ldots, x_j^n\}$ of observations and these observations of X_j are linked to the variables Y_j. The problem is to infer Y_j knowing the vector X_j. In general (but not always) we have to train the algorithm. In this case the algorithm must work on a given number of observations $\{Y_j, X_j\}_{1 \leqslant j \leqslant K}$ to build a model which will be used to perform the predictions. Building this model is equivalent to computing a function $\hat{Y} = f(X)$. Then, given an observation X_i the model can compute $\hat{Y}_i = f(X_i)$. When Y_i is known, we can compute the prediction error $\epsilon_i = Y_i - \hat{Y}_i = Y_i - f(X_i)$.

With the same situation, machine-learning can perform classification; in this case X belongs to a class Y_l for $l \in \{1, \ldots, p\}$ [1]. When we have an observation X_i, the prediction algorithm will have to predict the most probable class given the observation X_i. In this paper, the issue is a positioning problem, and thus it comes down to a regression problem.

A. k Nearest Neighbors (KNN)

K Nearest Neighbors (KNN) is one of the simplest machine-learning algorithms which was first described, to the best of our knowledge, in [7]. As previously stated, we have a given number of observations $\{Y_j, X_j\}_{1 \leqslant j \leqslant K}$ where X_j is usually a vector and Y_j is a real number.

Assuming that we have an observation X_i, we want to predict Y. The KNN algorithm must select the k nearest observations of X_i in $\{Y_j, X_j\}_{1 \leqslant j \leqslant K}$.

Let $i_1, \ldots, i_k$ be the k values which provide the k minimum values of the function

$$g(j) = d(X_j - X_i).$$

In other words $i_1, \ldots, i_k$ are the indexes of the k minimum values of $g(j) = d(X_j - X_i)$. These minimum values can be

[1] We can observe that regression and classification problems are very close.

equal if there are multiple values of X_j at the same distance from X_i.

We have at least the three possibilities for the distance, the most often used being the Euclidean distance.

$$
\begin{aligned}
d(X_j - X_i) &= \sqrt{\sum_{l=1}^{n} (x_l^j - x_l^i)^2} \quad \text{Euclidean} \\
d(X_j - X_i) &= \sum_{l=1}^{n} |x_l^j - x_l^i| \quad \text{Manhattan} \\
d(X_j - X_i) &= \left(\sum_{l=1}^{n} |x_l^j - x_l^i|^q \right)^{1/q} \quad \text{Minkowski}
\end{aligned}
$$

The value predicted for Y_i will be the mean value of the k values Y_j for the k nearest neighbors of x_i.

$$\hat{Y_i} = \frac{1}{k} \sum_{1}^{k} Y_{i_k}$$

B. Neural Networks

Neural networks take a given number of observations $\{Y_j, X_j\}_{1 \leqslant j \leqslant K}$ where X_j is usually a vector and Y_y is a real number. The idea is to build a network consisting of successive layers which combine the coordinates of X_j in successive layers to obtain the output which is the real number Y_y. The successive layers generally perform linear combination of the previous layer and the result in the neurones is obtained with an activation function which is usually a sigmoid function, see Figure III.1.

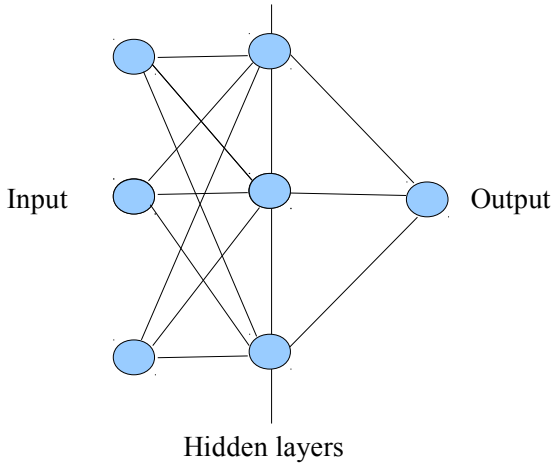

Fig. III.1. A neural network with, in this case, one hidden layer and three neurones.

To boost the performance of neural networks we have used ensemble neural networks. The idea is to randomly pick observations in the test set and create a neural network based on these observations. We can thus create many different sets of observations and for each set derive the associated neural network. Our final prediction network will be the average of these neural networks' predictions. The theory shows that the individual neural networks have a small bias and a high variance but the final prediction network will have a small bias and a small variance.

C. Random Forest

We still have a given number of observations $\{Y_j, X_j\}_{1 \leqslant j \leqslant K}$ where X is usually a vector and Y is a real number. The first step in a random forest scheme is to create a selection tree. Using the observations $\{Y_j, X_j\}_{1 \leqslant j \leqslant K}$ we build different sets using different splitting criteria which operate on the vectors $\{X_j\}_{1 \leqslant j \leqslant K}$. Each criterion allows the initial subset to be divided into two subsets. For instance, in Figure III.2 the criterion $X_j < A$ provides the first splitting of the observations. The following two criteria complete the selection tree which ends with four final leaf nodes.

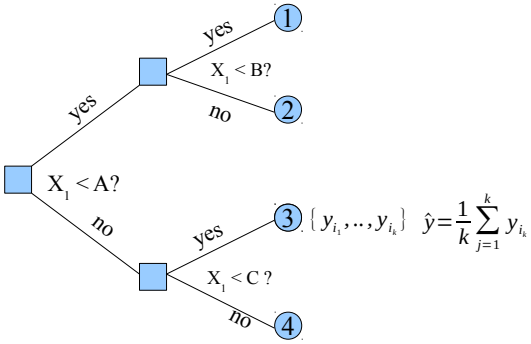

Fig. III.2. Regression tree .

Suppose now that we have a vector X_i and that we want to predict $\hat{Y_i}$. We will use the previous selection tree and determine in which final node the vector X_i is classified. Let us assume that X_i is classified in node 3 as are $X_{i_1}, \ldots, X_{i_k}$. In this case, the prediction of $\hat{Y_i}$ will simply be:

$$\hat{Y_i} = \frac{1}{k} \sum_{j=1}^{k} Y_{i_k}$$

The idea of Random Forest is to correct the error obtained in one selection tree by using the predictions of many independent trees and by using the average value predicted by all these trees. This Random Forest technique was first introduced in [8].

D. The Support Vector Machine regression technique

Generally the positions y_i and the related values x_i are known (thus we know $(y_i, x_i)_{1 \leqslant i \leqslant N}$) and we have to predict the positions using other values: $(x'_i)_{1 \leqslant i \leqslant N}$. We assume that

$$y_i = w^T \phi(x_i) + b \tag{III.1}$$

where w and b are two unknown vectors and $\phi(x)$ an unknown function of a vector x.

To solve these equations, we introduce the following convex optimization problem:

$$\text{minimize} \quad \frac{1}{2}||w||^2$$

$$\text{subject to } -\epsilon \leqslant w^T \phi(x_i) + b \leqslant \epsilon. \tag{III.2}$$

This problem assumes that the function given in III.1 can approximate the set of points that is given *i.e.* $(y_i, x_i)_{1 \leqslant i \leqslant N}$ with an accuracy of ϵ. Sometimes this is not possible and some errors must be accepted. In this case, slack variables which allow us to cope with impossible constraints, are introduced. This relaxation procedure uses a cost function. The convex problem then becomes :

$$\text{minimize} \quad \frac{1}{2}||w||^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*)$$

$$\text{subject to } -\epsilon - \xi_i^* \leqslant w^T \phi(x_i) + b \leqslant \epsilon + \xi_i \ \ with \ \ \xi_i^*, \xi_i > 0 \tag{III.3}$$

This problem can be solved by using Lagrange multipliers. The problem becomes:

$$\begin{aligned} L: \ = \ & \frac{1}{2}||w||^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) - \sum_{i=1}^{N}(\nu_i \xi_i + \nu_i^* \xi_i^*) \\ - \ & \sum_{i=1}^{N} \alpha_i (\epsilon + \xi_i - y_1 + w^T \phi(x_i) + b) \\ - \ & \sum_{i=1}^{N} \alpha_i^* (\epsilon + \xi_i^* + y_1 - w^T \phi(x_i) - b) \end{aligned}$$

where L is the Lagrangian and $\nu_i, \nu_i^*, \xi_i^*, \xi_i^*$ are the Lagrangian multipliers which are thus positive *i.e.* $\nu_i, \nu_i^*, \xi_i^*, \xi_i^* > 0$

We know that the minimum of L is attained when the partial derivatives are zero, thus:

$$\partial L/\partial b = \sum_{i=1}^{N}(\alpha_i - \alpha_i^*) = 0$$

$$\partial L/\partial w = w - \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)\phi(x_i) = 0$$

$$\partial L/\partial \xi_i^{(*)} = C - \alpha_i^{(*)} - \nu_i^{(*)} = 0$$

The substitution of these equations in the Lagrangian leads to the following problem:

$$\begin{aligned} maximize \quad & -\frac{1}{2}\sum_{i,j=1}^{N}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\phi(x_i)^T\phi(x_j) \\ & -\epsilon\sum_{i=1}^{N}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{N} y_i(\alpha_i - \alpha_i^*) \end{aligned}$$

subject to

$$\sum_{i,j=1}^{N}(\alpha_i - \alpha_i^*) = 0 \text{ and } \alpha_i, \alpha_i^* \in [0, C].$$

Thus we have:

$$w = \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)\phi(x_i)$$

and

$$f(x) = \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)\phi(x_i)^T\phi(x) + b$$

This formula is called the Support Vector expansion. The complexity of the function representation only depends on the dimensionality of the input space.

The rest of the analysis uses the Karush-Kuhn-Tucker (KKT) conditions. These conditions imply that at the solution the product between the constraints and the dual variable must vanish. In other words, we have

$$\begin{aligned} \alpha_i(\epsilon + \xi_i - y_i + w^T\phi(x_i) + b) = 0 \\ \alpha_i^*(\epsilon + \xi_i^* + y_i - w^T\phi(x_i) - b) = 0 \end{aligned} \tag{III.4}$$

and

$$\begin{aligned} (C - \alpha_i)\xi_i = 0 \\ (C - \alpha_i^*)\xi_i^* = 0 \end{aligned}$$

We can deduce that only the samples which do not satisfy the constraint of III.3 have $\alpha_i^{(*)} = C$. Moreover, since the two values of the right part of III.5 can not be simultaneously 0 then we have $\alpha_i \alpha_i^* = 0$. Thus after some observations we have :

$$\begin{aligned} \max(-\epsilon + y_1 - w^T\phi(x_i) | \alpha_i < C \text{ or } \alpha_i^* > 0) \leqslant b \leqslant \\ \min(-\epsilon + y_1 - w^T\phi(x_i) | \alpha_i < C \text{ or } \alpha_i^* > 0) \end{aligned}$$

This part is adapted from [9] a tutorial by Smola.

IV. Numerical results

Our numerical results are obtained on a straight road of length 600 m. The position on the road is given by $x \in [0, 600]$. We assume that we have three RoadSide Units (RSUs) located at $x = 0m$, $x = 305m$ and $x = 600m$.

Since we consider that there are no obstacles on the road to hinder free propagation, the signal strength received by the vehicles (or by the RSUs) depends solely on the distance between the vehicles and the RSUs. The power received is given by the following equation:

$$P = \frac{P_0}{r^\beta} \quad with \quad \beta \in [2, 4]$$

We measure the power received in dB thus

$$P^{dB} = 10\frac{log(P)}{log(10)}.$$

Moreover, errors in the measurements are taken into account; we assume a Gaussian noise of zero mean and with a variance 0.05. This can also be interpreted by a log-normal fading which would affect the reception. This assumption is realistic when the transmissions between the vehicles and the RSUs are in line of sight. In the second analysis we assume a Raleigh fading of rate $\mu = 1$. This corresponds to the case where there are multi-path links between sources and destinations as, for example, in built up areas in a town.

The data base is obtained by 20 different measurements at each location of the vehicle, the locations being 15 meters apart. Thus the data consist of 780 sets with three different powers, each of them corresponding to the power received by the three roadside units respectively.

Even if it were possible to do otherwise, for the sake of simplicity we assume that the vehicles send beacons which are received by the three roadside units. The RSUs fuse these data and perform the machine-learning process. The location of the vehicles having been established, it can then be sent to them by one of the RSUs.

For the KNN algorithm, we use the data set directly derived from the power measurements in dB; we do not perform any data processing before using the KNN algorithm. The code of KNN is found in the R software [10] and in its KNN library. We use the Euclidean distance.

For the Neural Network we use a library coded in Visual C++. The neural network has one hidden layer with three neurones. The activation function used is a sigmoid. We create an ensemble of 50 neural networks using a bagging technique and the final prediction is that of the average of the 50 neural networks previously obtained.

For the Random Forest algorithm, we use the data set directly derived from the power measurements in dB; we do not process the data before using the algorithm. We use the Random Forest library found in the R software [10].

For the Support Vector Machine, the libsvm library [11] is used. The data set (powers in dB) is not directly processed. A linear transformation of these powers is performed so that the minimum power becomes 0 and the maximum power 1. The following values for the parameters are used: $C = 10$, $\epsilon = 10^{-6}$ and an exponential kernel. This means that we have to significantly increase the penalization of not respecting the bounds for the estimation since the default value for C is 1.

A. Direct link and log-normal fading

The comparison between KNN, NN, Random Forest and Support Vector Machine is presented in Figure IV.1. We observe that except for KNN the position error remains in the interval $[-20m, 20m]$, which shows that when have a direct link, the machine-learning techniques offer reasonably good predictions. We also observe that the NN and RF techniques seem to perform the best and apparently there is no clear segment on the road where the prediction is better. We nonetheless observe a notable degradation of the prediction close to $x = 300m$ for the KNN and the RF techniques.

Table I proposes a quantitative evaluation of the four methods with the mean absolute error and the root mean square deviation. From these results we note the NN approach provides the best performances with a mean absolute error of $3.3m$ and a root mean square deviation of $4.7m$ followed by the RF technique with an absolute error of $5.2m$ and a root mean square deviation of $7.1m$. We then have the KNN scheme with an absolute error of $7.3m$ and a root mean square deviation of $10.8m$ and the least effective scheme is SVM with an absolute error of $9.4m$ and a root mean square deviation of $11.1m$

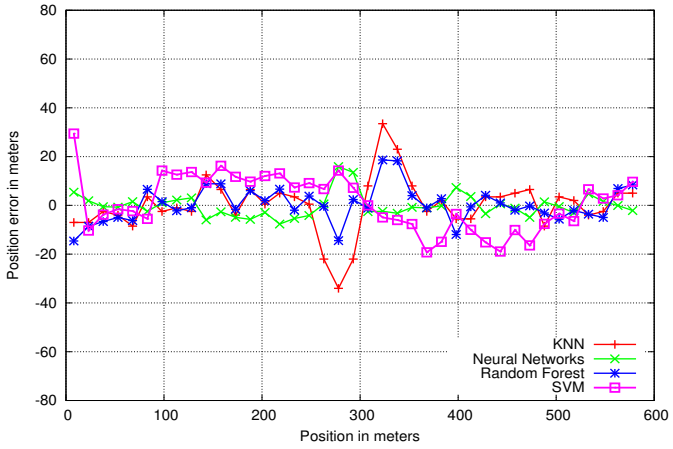

Fig. IV.1. Position errors versus position $x \in [0, 600m]$ on the roads with the different machine-learning techniques $\sigma = 0.05$.

TABLE I
Mean error and root mean square deviation versus prediction techniques (direct path propagation)

KNN (m, σ)	NN (m, σ)	RF (m, σ)	SVM (m, σ)
(7.31 , 10.8)	(3.32 , 4.67)	(5.24 , 7.06)	(9.44 , 11.14)

B. Direct link and log-normal fading but measurements only every 30m

Here we perform the same comparison as in the previous section but we only have training data every $30m$ instead of every $15m$ as in the previous section. The predictions of the four algorithms remains acceptable and the ranking is still: first Neural Network, second Random Forest, third KNN and fourth SVM. Table IV provides the quantitative results; NN offers the best estimation with an absolute mean error of $4.7m$ with an RMS of $7.8m$, then comes RF with an absolute mean error of $5.85m$ and an RMS of $7.3m$. The two last places are for KNN: absolute mean error of $8.8m$ and an RMS of $13.8m$. followed by SVM: absolute mean error of $10.1m$ and an RMS of $11.3m$.

The degradation of precision of the localization is roughly 50% in terms of absolute error for NN and RF and there is no significant degradation for KNN and SVM.

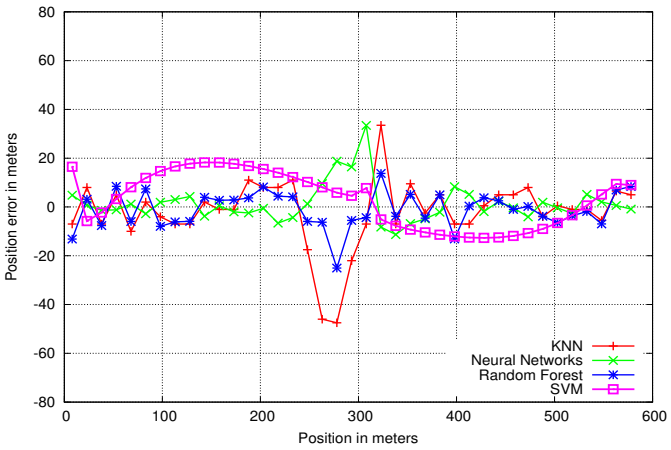

Fig. IV.2. Position errors versus position $x \in [0, 600m]$ on the roads with the different machine-learning techniques $\sigma = 0.05$. Training data measurements available only every 30m

TABLE II
MEAN ERROR AND ROOT MEAN SQUARE DEVIATION VERSUS PREDICTION TECHNIQUES (DIRECT PATH PROPAGATION BUT MEASUREMENTS ONLY EVERY 30M)

$KNN\ (m, \sigma)$	$NN\ (m, \sigma)$	RF (m, σ)	SVM (m, σ)
(8.81 , 13.79)	(4.75 , 7.78)	(5.85 , 7.33)	(10.13 , 11.27)

C. Direct link and log-normal fading but no measurement in the segment [30m,105m]

Here we study the performance of our algorithms when we have no training data for a given section of the road for $x \in [30m, 105m]$. We observe (except for NN) that the algorithms exhibit larger estimation errors in the segment where there is no training data available and the mean absolute error also increases. We still have the following ranking of the four algorithms: NN, RD, SVM and KNN. We observe that the least effective scheme in this scenario is KNN but we have not changed the number of nearest neighbors (which might be considered unfair).

Table III provides a qualitative analysis of the scenario with no data for $x \in [30m, 105m]$. For the NN algorithm the performance degradation is around 30% whereas the degradation is around 60% for the RF algorithm. There is almost no degradation for SVM and the degradation is around 130% for KNN.

A significant loss in the training data leads to a significant degradation in the estimation of the position, except for NN.

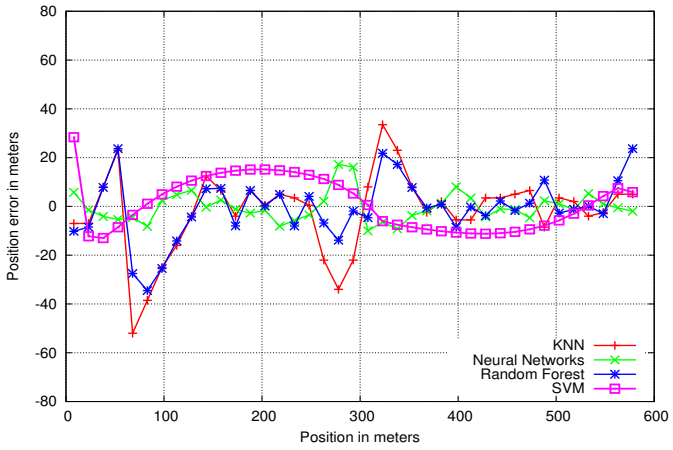

Fig. IV.3. Position errors versus position $x \in [0, 600m]$ on the roads with the different machine-learning techniques $\sigma = 0.05$. No training data available in [30m,105m]

TABLE III
MEAN ERROR AND ROOT MEAN SQUARE DEVIATION VERSUS PREDICTION TECHNIQUES (DIRECT PATH PROPAGATION BUT NO TRAINING DATA AVAILABLE IN [30M,105M])

$KNN\ (m, \sigma)$	$NN\ (m, \sigma)$	RF (m, σ)	SVM (m, σ)
(18.35 , 23.73)	(4.33 , 5.78)	(8.69 , 12.17)	(9,23, 10.60)

D. No prominent path and Rayleigh fading (no direct path Rayleigh fading)

In the following, we present the results of our four algorithms when the power received is more affected by the fading. A Rayleigh fading (of rate 1) is assumed which means that there is no prominent direct path between the vehicle and the roadside units. The power received consists in a random combination of many independent paths. In these conditions, the predictions, without filtering the measurement, lead to really poor results. Thus they are not included in our presentation.

The comparison between KNN, NN, Random Forest and Support Vector Machine is presented in Figure IV.4. We observe that,except for KNN, the position error remains in the interval $[-60m, 60m]$, which shows that when have no direct link, the machine-learning techniques offer only average predictions.

Table IV proposes a quantitative evaluation of the four methods with the mean absolute error and the root mean square

deviation. From these results we note the NN and SVM approaches provide the best performances with mean absolute errors of respectively $17.63m$ and of $17.45m$ and with root mean square deviations of respectively $23.08m$ and $23.51m$. These techniques are followed by the RF technique with an absolute error of $26.9m$ and a root mean square deviation of $37.1m$. The least effective scheme is KNN with an absolute error of $30.30m$ and a root mean square deviation of $35.63m$

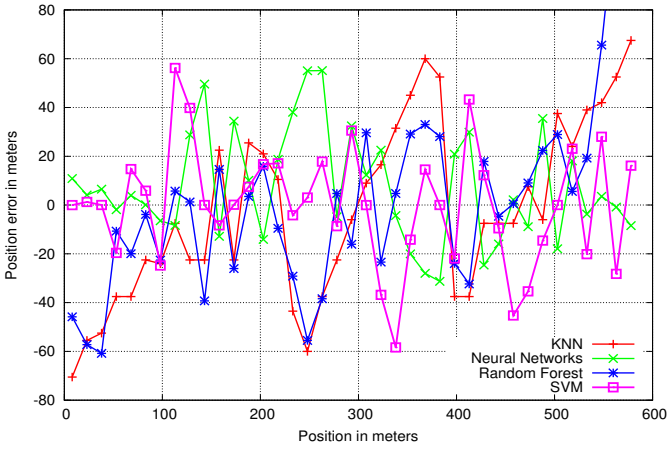

Fig. IV.4. Position errors versus position $x \in [0, 600m]$ on the roads with the different machine-learning techniques with Rayleigh fading .

TABLE IV
MEAN ERROR AND ROOT MEAN SQUARE DEVIATION VERSUS PREDICTION TECHNIQUES (NO DIRECT PATH PROPAGATION)

KNN (m, σ)	NN (m, σ)	RF (m, σ)	SVM (m, σ)
(30.30 , 35.63)	(17.63 , 23,08)	(26,94 , 37,15)	(17,45 , 23.51)

V. CONCLUSION

In this paper, we compare four machine-learning techniques to predict the position of a vehicle using the reception power of packets sent to fixed nodes whose positions are precisely known.

We have studied the KNN technique, the Neural Network technique, the Random Forest technique and the Support Vector Machine technique. The simplest method is the KNN technique: in the data set the scheme selects the k closest samples of the actual measurement. The neural scheme we have tested in this paper consists of one hidden layer with three neurones. To boost this technique we use an ensemble neural network with 50 elements built with a bagging algorithm. In the Random Forest scheme, we use a classification tree to generate different classes according to a random classification tree. The location in each class is assumed to be the average location of the points in this class. The tree is then used for the prediction; the location predicted being that of the training samples at the same leaf of the random trees. The Support Vector Machine is an approximation technique which usually uses kernels as base functions. The main goal is to maintain, as far as possible, the samples and their approximations with a bounded error as much as possible. In general, the base functions are exponential functions.

The numerical experiments presented in this paper demonstrate that a precise prediction can only be obtained when there is a main direct path of propagation. The prediction is altered when the training is incomplete or less precise but the precision remains acceptable. In contrast, with Rayleigh fading, the accuracy obtained is much less striking. We observe that the Neural Network is nearly always the best approach. With a direct path the ranking is: Neural Network, Random Forest, KNN and SVM except in the case when we have no measurement in $[30m, 105m]$ where the ranking is Neural Network, Random Forest, SVM and KNN. When there is no direct path, the ranking is SVM, NN, RF and KNN but the difference in performance between SVM and NN is small.

REFERENCES

[1] *Task Group p. IEEE 802.11p*, Wireless Access in Vehicular Environments (WAVE) Draft Standard, 2007.

[2] ETSI EN 302 637-2, "Intelligent Transport Systems (ITS) - Vehicular Communications - Basic Set of Applications - Part 2 : Specification of Cooperative Awareness Basic Service," *History*, vol. 1, pp. 1–44, 2014.

[3] ETSI EN 302 637-3, "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service," vol. 2, pp. 1–73, 2014.

[4] M. Porretta, P. Nepa, G. Manara, and F. Giannetti, "Location, location, location," *IEEE Vehicular Technology Magazine*, vol. 3, 2008.

[5] B. C. Liu and K. H. Lin, "Sssd-based mobile positioning: On the accuracy improvement issues in distance and location estimations," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 3, pp. 1245–1254, March 2009.

[6] S. A. E. Mohamed, "Why the accuracy of the received signal strengths as a positioning technique was not accurate?" *International Journal of Wireless and Mobile Networks (IJWMN)*, vol. 3, no. 3, pp. 69–82, June 2011.

[7] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/00031305.1992.10475879

[8] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001. [Online]. Available: https://doi.org/10.1023/A:1010933404324

[9] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.

[10] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2013. [Online]. Available: http://www.R-project.org/

[11] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 27:1–27:27, May 2011. [Online]. Available: http://doi.acm.org/10.1145/1961189.1961199