# WIMAX Traffic Forecasting based on Neural Networks in Wavelet Domain

Ion Railean*,‡, Cristina Stolojescu*,†, Sorin Moga*, Philippe Lenca*
*Institut Telecom; Telecom Bretagne, UMR CNRS 3192
Lab-STICC; Université europèenne de Bretagne, France
Email: firstname.lastname@telecom-bretagne.eu
†"Politehnica University" of Timisoara, Romania
Email: firstname.lastname@etc.upt.ro
‡Tehnical University of Cluj Napoca, Romania

***Abstract*—In this article we propose an approach for predicting traffic time series based on the association of the Stationary Wavelet Transform (SWT) with Artificial Neural Networks (ANN). We focused on comparing the quality of forecasting obtained using different configurations of the ANN. We tested our different configurations using real traffic data recorded at each base station that belongs to a WiMAX Network developed by Alcatel. We compared our approach with previously forecasting models using ANNs and showed the performance of our neural network configuration.**

## I. Introduction

Network traffic prediction plays a fundamental role in characterizing the network performance and it is of significant interests in many network applications, such as admission control or network management. For a WIMAW network, and globaly for a radio network, predicting the future traffic level is mandatory in order to keep a satisfactory quality of services. The decision of changing the network architecture (i.e. add new base stations) is essentialy based on prediction results. Models that accurately catch the characteristics of actual traffic are useful for analysis and simulation, and they help to understand the network dynamics and to design and control the network. The main idea of the traffic forecasting is to precisely predict traffic in the future, considering the measured traffic history. The choice of the prediction method is based on the prediction interval, prediction error and computational cost. Mining time series data is one of the most chalanging problem in data mining research [28] and in order to come with a suitable conclusion regarding what prediction technique to use for this purpose, different types of forecasting methods have been studied.

The traditional approaches to time series forecasting assume that the time series are issued from linear processes, but they may be totally inappropriate if the underlying mechanism is nonlinear [26]. One of the models is based on Box-Jenkins methodology which is used to build the time series model in a sequence of steps which are repeated until the optimum model is achieved. Another class of models uses the structural state space methods and can be used to predict the stationary, trend, seasonal, and cyclical data. These methods capture the observations as a sum of separate components (such as trend and seasonality). Between all the forecasting models, artificial neural networks (ANNs) have been shown to produce better results [4], [17] and [12]. In [8] the performance and the computational complexity of ANNs are compared with the ones obtained using ARIMA and fractional ARIMA (FARIMA) predictors, Wavelet based predictors and ANNs. The results of this study show the significant advantages for the ANN technique. In [5], the advantage of the ANN over traditional rule-based systems is proved. The authors of [25], [6] and [27] propose a time delayed neural network (TDNN).

Using Wavelet Transform The forecasting accuracy by using Wavelet Transform is described in [19]. The paper presents a forecasting technique for forward energy prices, one day ahead. The results demonstrate that the use of Wavelet Transform as a pre-processing procedure of forecasting data improves the performance of prediction techniques.

The originality of our proposed forecasting model is based on time series decomposition in wavelet domain and the use of ANNs in the transform domain. This model was developped as result of a Alcatel-Lucent study [9]. The aim of this paper is to compare different configurations of the ANN in order to find the highest prediction performance.

The rest of the paper is organized as follows: Section 2 provides a presentation of the forecasting framework including some theoretical considerations regarding the wavelet transform, the ANNs and the genetic algorithms. In Section 3 we present the configuration of the artificial neural network, and the ANN optimization. The next section describes the measures used for the performance evaluation. The results of our experimental work and the description of the used data base are presented in section 5. Finally, the last section is dedicated to conclusions and further work.

## II. Forecasting framework

The main idea of the proposed prediction method using the wavelet transform is to decompose the original data (referred also as the time-series signal in this article) into a range of frequency scales and then to apply the forecasting method to these individual components. It implies several steps as presented in Fig. 1:

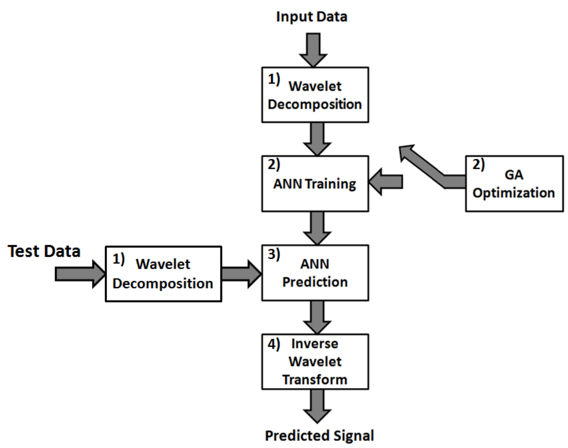


Fig. 1. The basic four steps of the forecasting framework

1) Use the Stationary Wavelet Transform to decompose the data for input and for test.
2) Apply the Artificial Neural Networks for each level of decomposition obtained from the input and build the forecasting model. Choose between having or not the Genetic Algorithm Optimization for the ANNs.
3) Use the decomposed signal and the obtained model in order to predict each decomposition level of the future forecasted signal.
4) Use the Inverse Stationary Wavelet Transform in order to obtain the final predicted signal.

### A. The Wavelet Transform

Wavelets are mathematical functions that cut up data into several frequency components, then process them at different scales or resolutions. The multi-resolution analysis (MRA) is a signal processing technique that takes into consideration the signal's representation at multiple time resolutions. At each temporal resolution two categories of coefficients are obtained: approximation and detail coefficients. Usually, the MRA is implemented based on the algorithm proposed by Stephane Mallat [16], which corresponds to the computation of the Discrete Wavelet Transform (DWT). The disadvantage of this algorithm is the decreasing of the coefficient sequences length with the increasing of the iteration index due to the utilization of the decimators. Another way to implement a MRA is the use of the  trous methodology [20], which corresponds to the computation of the Stationary Wavelet Transform (SWT). In this case the utilization of decimators is avoided, but at each iteration different low-pass and high-pass filters are used. In order to assemble back these components into the original signal without loss of information, it is used the process called reconstruction, or synthesis. The reconstruction is done using the Inverse Stationary Wavelet Transform (ISWT).

### B. Artificial Neural Networks

An Artificial Neural Network [11] is a mathematical non-linear model which has been very used recently in the area of forecasting. It is composed of interconnected simple elements, called artificial neurons. An ANN is characterized by three characteristics: the architecture (interconnection of neural units), the learning or training algorithm (method for determining the weights of the connections), and the activation function (produces the output based on the input values received by a node).

There are different architectures of ANNs, which consequently require different types of algorithms. The two most important types of ANNs are Feed-forward and Recurrent network [11]. A feed-forward neural network has its neurons organized in a layered structure. Each layer consists of units which receive their input from the units situated on a layer directly below and send their output to the units from a layer directly above. There are no connections within a layer. By contrast, Recurrent neural networks are characterized by the fact that they contain feedback connections (at least one feedback loop [11]) and they take into consideration the dynamical properties of the network [15]. A recurrent network can be formed only by one layer of units, each unit feeding its output signal back to the inputs of all the other neurons.

The ANN has to be configured in such a way that the application of a set of inputs produces the desired set of outputs [1]. This can be done by changing the weights between the neurons from different layers. There are various methods to set these weights. One way is to set them explicitly, using a priori knowledge. Another way is to 'train' the ANN by feeding it teaching patterns and letting it change its weights according to some learning rule. Training the network is time consuming. It usually learns after several epochs, depending on how large the network is. Thus, a large network requires more training time compared to a smaller one. Another way to set the weights is to apply a genetic algorithm [2]. The optimization function implemented by GA will consist in minimizing the output error.

### C. Genetic Algorithms

A genetic algorithm (GA) represents a search technique for optimization and machine learning applications, based on natural selection [2]. It includes a set of individual elements represented by vectors (the population) and a set of operators defined on population. At each step, the GA selects individuals randomly from the current population to be parents and uses them to produce the children for the next generation. Over successive generations, the population 'evolves' towards an optimal solution. There are several steps in order to apply a GA:

- Encoding technique: gene, chromosome
- Initialization procedure: creation
- Evaluation function: environment
- Selection of parents: reproduction
- Genetic operators: mutation, recombination
- Parameter settings: practice and art

A GA selects the subsets (usually pairs) of solutions from a population (parents), to combine them to produce new solutions (children). The rules of combination are based on the genetic notion of crossover, which consists of interchanging solution values of particular variables, together with occasional operations such as random value changes, called mutations. The children produced by the mating of parents that pass a survivability test, are then available to be chosen as parents for the next generation.

## III. Configuration of the Artificial Neural Network

In order to follow better our approaches in designing the neural networks, we present the detailed scheme of the ANNs' application shown in figure 2. The first step is to split the data in training and testing data sets. The next step is the MRA pre-processing of both training and testing data sets (second step in 2). The $n^{th}$ level of decomposition depends on the length of the input data, so, according to our block diagram it depends on number of samples per day we can choose. In [18] it is pointed out that in order to be able to apply the SWT for a discrete signal, then the signal must divide to $2^n$ if a decomposition at level *n* is needed. The initial data has been provided with 96 samples/day (description of the data will be done in section V), resulting in a level of decomposition *n*=5. However, according to [22] and [9], the WiMAX network traffic exhibits some periodicities (of 12 hours, 24 hours,...). These periodicities are better noticed if we have a sampling rate at each 45 or 90 minutes. It results 32 or 16 samples/day. In this case, our maximum level of decomposition is 4, or 3 respectively. Therefore, we have to train and model three, four, or five ANNs.

This type of modeling was applied for day and week prediction. For one day prediction we propose two approaches:

1) The first approach, which was named by us "*Similar Days Selection*", consists in the selection of the days which are similar to the one we want to predict. For example, if we want to predict how the traffic behaves on Wednesday, then for the preprocessing phase we take only the data corresponding to all Wednesdays during the first seven weeks. It results (in the training phase) that we have six days for future ANNs inputs and one day for ANNs targets. The advantage in this technique is that usually the network user's behavior is modeled during certain week days.
2) The second approach for day prediction, named "*All Days Selection*", is to take into consideration all days until the day we want to forecast.

In [10], in a research concerning the forecast of an entire week, it is shown that ANN outperforms the linear models. This is one of the reasons why we consider the prediction of one or several weeks ahead. In this case we propose three methods:

- The first method, called "*No Sliding*", is the simplest one: we train the ANN once for each decomposition level. For inputs, we have the first (*n*-2*k*) weeks, where *n* is the total number of weeks, and *k* is the number of weeks we want to forecast. The target consists of the data taken from (*n*-2*k*+1) : (*n*-*k*). The output signal is compared to the real data of the last *k* weeks. For example, if we want to predict the last two weeks (weeks 7 and 8), then in the training process we take the data from the weeks 5 and 6, and we leave the data from weeks 1-4 for ANNs' inputs.
- The second method uses sliding with retraining the network with the real data. The method is called "*Known Sliding*". The entire data is divided into smaller parts. Each of this sequences will predict a small part from the final forecasted data. The information for neural networks retraining is always taken from the real data.
- In the last method known as "*UnKnown Sliding*", the information used for the next simulation and retraining is taken not from the original data, but from the previously predicted one.

In order to implement an artificial neural network, first of all we had to choose its model. According to [26] feed-forward neural network is relatively accurate in forecasting, despite being quite simple and easy to use. Also, during our model implementation, the recurrent network forecast performance was lower than that of the feed-forward model. These are the reasons why feed-forward ANNs were used in our forecasting technique.

Further designing of the ANN implies the establishment of the number of layers and the number of neurons in each layer. These aspects are very important if we want to minimize the generalization error, the learning time, and the ANN dimension. In [13] is pointed out the fact that the choice of the number of layers is made knowing that one hidden layer network is able to approximate most of the nonlinear functions demanded by practice. Considering this observation we chose three layers ANNs. Concerning the dimension of each layer the situation is as follows: input and output layers are imposed by the problem to be solved, while the dimension of the hidden layer is essential for efficiency of the network.

*1) Input and output layer:* In the case of *one day prediction, Similar Days Selection* model, we want to predict a single vector (data from one day) having a length of 96, 32, or 16 samples. This is why we used only one neuron for the output.

Regarding the usage of the data for the input layer, we used the concept of the inputs of a time delayed neural network (TDNN) described in [25], [6] and [27]. The idea of such a network is to use overlapping frames in time. For this overlapping we had several options. The first option was to set this overlapping to zero, so, we made a temporal synchronization during the whole day (24 hours shifting). One of the methods to achieve this was by taking an entire day from midnight till midnight as input of the ANNs (data collected during a morning period are used to learn and forecast future morning period, etc.). Another option in choosing the overlapping step for our data, was related to the human's behavior and activites during a day (e.g. eight hours of sleep, eight hours of work, and the remaining eight hours for rest).

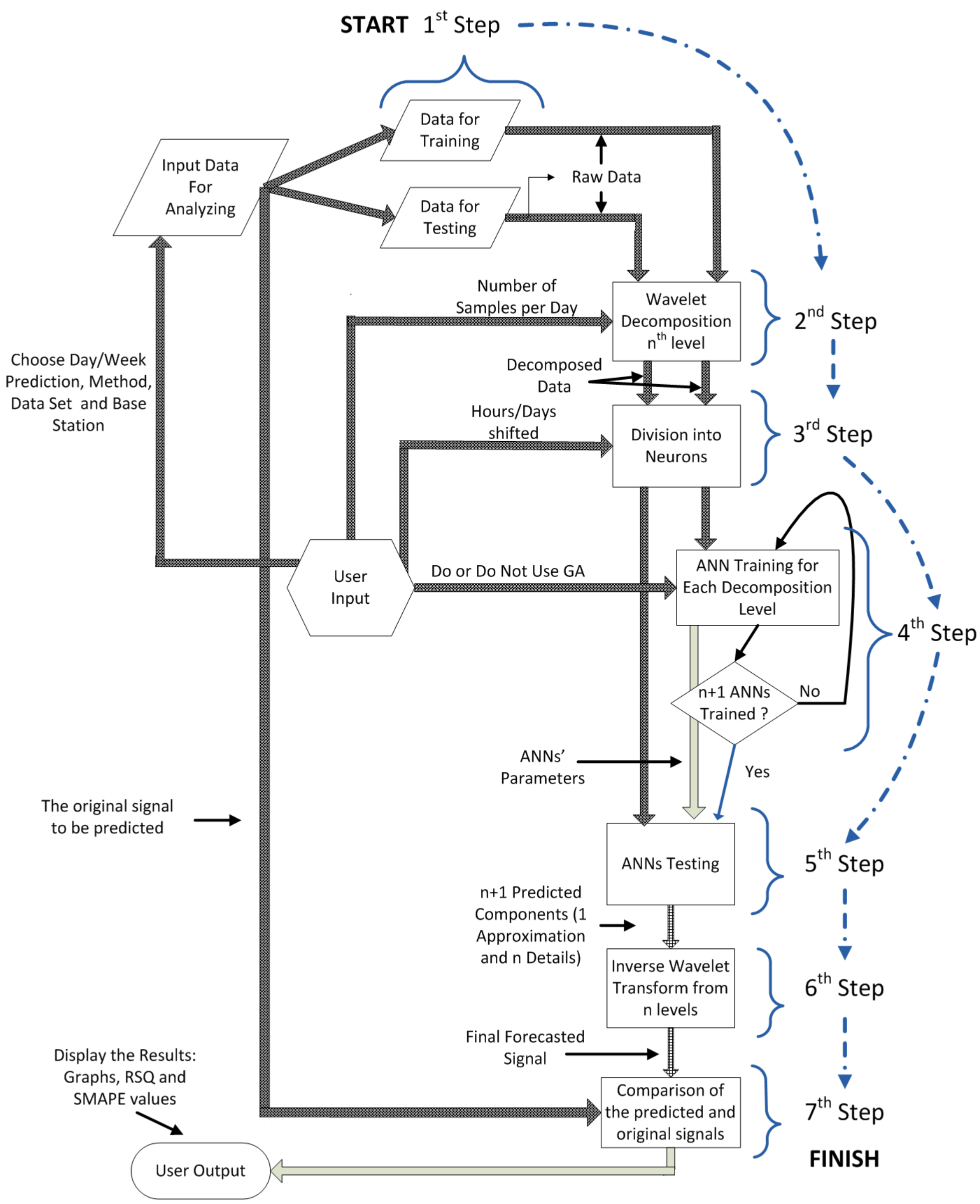


Fig. 2. Block Diagram of the ANN Modeling

Based on these aspects, the data for neurons had been taken by making a shift of 8 hours time interval, but the length of each of them still remains of 24 hours. So, we had 2/3 of similar data for two adjacent neurons. Anyway, in [21], it is described that a recent survey on the range of shift systems that are currently operating, found that about 1/3 of continuous systems now involve 12 hour shifts. The same survey noted that there was a range of different types of 12 hour systems. The organizations that participated at this research, were manufacturing companies (steel, chemicals, aluminum, oil, chipboard, food, glass fibre) along with one engineering company. Based on this article, we made a 12 hours shift between ANN's neurons. It resulted that the total number of input neurons was equal to the number of subsequences which could be obtained when taking the data with an overlapping of 8, 12 or 24 hours. It should be also pointed out that the initial data is the concatenation of information over entire 6 weeks.

It can be easily obtained the following formula for neurons number calculation:

$$NrI = (NrD - 1) \cdot \frac{24h}{NrHShift} + 1 \qquad (1)$$

where *NrI* is the number of input neurons, *NrD* represents the number of days taken into consideration, and *NrHShift* expresses the number of shifted hours. So, in case of 24 hours shifting we had 7 neurons for input, for 12 hours delay there were 13 input neurons, while for 8 hours we obtained 19 input neurons respectively.

Concerning the *All Days Selection* model for one day prediction, our approach had the same basic ideas as the first one. The main difference was the array of data received from the SWT prepared for neural network's inputs. For no overlapping (shifting of 24 hours) in this case, by using (5), the number of input neurons was 44. For 12 hours we obtained 87 neurons, while for 8 hours shifting the result was 130 neurons.

In the case of *one week prediction*, we had also a single output neuron. For the input layer we applied again the ideas from a TDNN. First, we tried to understand the behavior of different companies regarding the days of working, or resting, given to their employees. In [21] is pointed out that there is a form of system involving work with four shifts system - for example, two days shifts, followed immediately by two night shifts, followed by several rest days. This type of day working is usually popular in medicine, and restaurant business. According to the information just described, we implemented a variable day shifting at the inputs of our ANN also. Based on the same analysis as in the case of day prediction, we obtained the next formula for calculation of the number of input neurons:

$$NrI = \left\lfloor (NrW - 1) \cdot \frac{7[Days]}{NrDShift} \right\rfloor + 1 \qquad (2)$$

where $\lfloor x \rfloor$ is the *floor function*, also called the greatest integer function or integer value, and gives the largest integer less than or equal to $x$, *NrW* expresses the number of weeks, *NrDShift* represents the number of shifted days used in analysis. For example, in the case of two days shifting, number of input neurons was 18.

*2) Hidden layer:* If the number of input and output neurons were kind of 'decided' by the data we have, the hidden layer stood completely at our choice of designing.

In [13], it is said that the network dimension must satisfy at least two criteria: the network must be able to learn the input data and the network must be able to generalize for similar input data that was not used in training set. The accomplishment degree of these requirements depends on the network complexity, training data set and number of iterations for training. After simulations and tests, we arrived at the conclusion that the best value for the number of neurons in the hidden layer is 3 for one day forecasting, and 12 for one week forecasting as shown in figure 3 and figure 4. In the case of one day prediction the number of simulations and results for each choice regarding the number of neurons was 67 (the number of base stations belonging to the WiMax network) multiplied by the total number of days in a week (7 days). We used 8 hours shifting. The values of SMAPE presented in 3 represent the mean values obtained after these simulations. Concerning the case of week prediction the number of simulations and results for each dimension of the hidden layer was 67. We used one week ahead forecasting and one day shifting. Each value of SMAPE in 4 represent the mean of the 67 obtained results. By increasing the size of the hidden layer, the network was not able to make a good generalization, and was overlearning itself, meaning that it extracted too much information from the individual cases forgetting the relevant information of the general case.

Another problem that stays in the ANN design consists in choosing the training periods and the performance (Mean Squared Error (MSE)). These two parameters are related to each other, meaning that more periods will usually result in a better performance. It is considered that during training, the MSE between the output signal and the target should be about 5%, [7]. So, we set the limits for MSE to be 0,05.

Another important parameter in ANN design, is the decision concerning the training function to be used. Matlab has several training functions such as: Batch Gradient Descent (traingd), Batch Gradient Descent with Momentum (traingdm), Back-propagation training with an adaptive learning rate (traingda) or a combination between Adaptive Learning rate with Momentum training (traingdx). After several tests, we observed that "traingd" gets blocked very fast and it is not able to a further converging of training (it usually stopped at a MSE value of 0.74). "traingdm" even if it is said that it does not get stocked in a local minimum, in our case we had several unsuccessfull simulations regarding this aspect (the ANN did also stop at a performance value of 0.95 and was not able to converge further). Both "traingda" and "traingdx" managed to train the network, but "traingda" was a little bit slower then "traingdx": for example, after 900 iterations of training, "traingda" reached 0.573 performance value, while "traingdx" had a MSE of 0.532. So, we used the Adaptive Learning Rate with Momentum Training function for our networks.

*Optimization*

The next step in our prediction model, was to see how can we optimize our ANNs using Genetic Algorithms. We applied the GA in order to find the optimal weights between the input and hidden layer. Because of the number of links between these two layers, we applied GA only for the *Similar Days Selection* model. Also, in order to diminish the time needed for processing and the number of variables needed (represented by genes), the hidden layer contained just 2 neurons instead of 3. Matlab finds difficult to search for more than 7 - 8 variables using genetic algorithms. When we applied GA for the *All Days Selection* model of prediction and for one week prediction model, then according to the next equation:

$$[\text{NrLinks}] = [\text{NrInputNeurons}] \cdot [\text{NrHiddenNeurons}] \qquad (3)$$

the number of links were $44 \cdot 3 = 132$, respective $6 \cdot 12 = 72$ links, implying the same number of variables. While for the *Similar Days Selection* model, there were $7 \cdot 2 = 14$ links for 24 hours shifting.

The designing of training the ANN using the Genetic Algorithms was as follows:

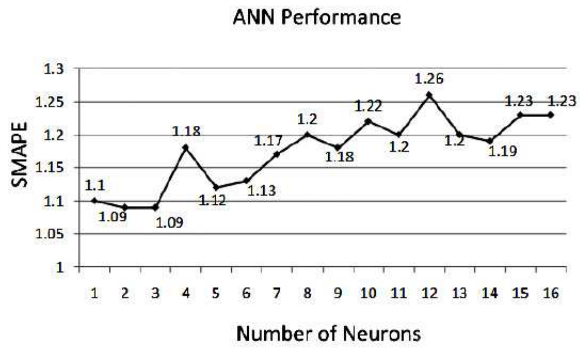


Fig. 3. SMAPE values as a function of neurons number in the hidden layer, one day prediction

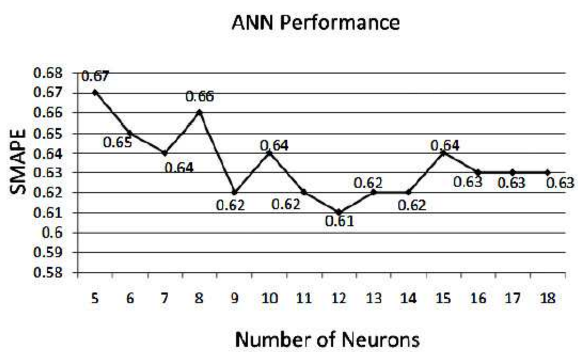


Fig. 4. SMAPE values as a function of neurons number in the hidden layer, one week prediction

- each individual contains a set of weights for all the links between layers
- each gene represents a single weight
- we had a population size of 100 individuals, meaning 100 different possibilities at each generation for the network
- the number of generations is 100: less generations resulted in not finding an acceptable solution for our problem, while more generations resulted in a longer time processing. However, above this value, we didn't manage to observe better performance of the final results
- the fitness function representing the summation between the two training data sets, is calculated as follows:

$$F = \frac{1}{[N]}\sum_{i=1}^{[N]}(x_i^{f_1} - x_i^{o_1}) + \frac{1}{[N]}\sum_{i=1}^{[N}(x_i^{f_2} - x_i^{o_2}),$$

where N is the number of samples, $x_i^{o_1}$and $x_i^{o_2}$ represent certain level decomposition of the original signal used for inputs, while $x_i^{f_1}$ and $x_i^{f_2}$ are the output targeting signals for the two data sets.

We had eight weeks of data. The *Similar Days Selection* model uses one day from each week. In order to use GA, we needed two different signals for training. For the calculation of the first part of the fitness function, we used the samples from wavelet transform of the desired day from each of the weeks $1^{st}$ till $5^{th}$ at the input of the ANNs. We used the wavelet transform of the day from week number 6 as a target data. The second part of the fitness function, was calculated by applying at the network's inputs the data from the second training signal: the wavelet decomposition of the wanted days from weeks $2^{nd}$ till $6^{th}$ as input, and the wavelet decomposition of the day belonging to the $7^{th}$ week as target. So, during one generation of a given individual, and supposing ten links between the first two layers, we have the next steps:

- Take those ten genes of the individual and apply them as weights to the network
- Input data from the wavelet transform of the first five weeks, compare the output with the SWT of the day from the $6^{th}$ week, retain the first part of the fitness function
- During the same generation, using the same weights, apply the SWT of the days starting with the second week till the $6^{th}$, compare the output with SWT of the day from the $7^{th}$ week, retain the second part of the fitness function
- Compute the final value for the fitness function
- Apply mutation and crossover to these weights (genes), use their new values (children) for the next generation

Using this GA method for training ensures us a better approximation of the signals. It is like training the same network with two different sets of data, kind of ”*double-training*”, which cannot be accomplished by using the Neural Network Toolbox from Matlab.

## IV. Measuring accuracy

The predictive ability of the forecasting model was evaluated in terms of the following well-known evaluation criteria: Symmetrical Mean Absolute Percentage Error (SMAPE) and R-Square (RSQ).

- Symmetric Mean Absolute Percent Error (SMAPE): calculates the symmetric absolute error in percent between the actual *X* and the forecast *F* across all observations *t* of the test set of size *n* for each time series *s.*

$$SMAPE = \frac{1}{n}\sum_{t=1}^{n}\frac{|X_t - F_t|}{(X_t + F_t)/2} \tag{4}$$

  In the ideal case the value of RSQ is 1, while SMAPE must be 0.
- R-Square (RSQ): The coefficient of determination $R^2$, in statistics, is the proportion of variability in a data set that is accounted for by a statistical model. In this definition, the term *variability* is defined as the sum of squares. A version for its calculation is:

$$R^2 = \frac{SS_R}{SS_T} \tag{5}$$

  where:

$$SS_T = \sum_t (X_t - \overline{X_t})^2 \tag{6}$$

$$SS_R = \sum_t (F_t - \overline{F_t})^2 \tag{7}$$

in which $X_t$, $F_t$ are the original data values and modeled values (predicted) respectively, while $\overline{X_t}$ and $\overline{F_t}$ are the means of the observed data and modeled (predicted) values, respectively. $SS_T$ is the total sum of squares, $SS_R$ is the regression sum of squares.

- Mean absolute error(MAE): represents the average absolute error value. The mean absolute error (MAE) is given by:

$$MAE = \frac{1}{T}\sum_{1}^{T}|F_t - X_t| \tag{8}$$

where $F_t$ is the prediction and $Y_t$ the true value.

- Root Mean Square Error(RMSE): it measures the differences between the values predicted by the model and the values actually observed from the time-series being modeled or estimated.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{\sum_1^T (X_t - F_t)^2}{\sum_1^T (X_t - \overline{X}_t)^2}} \tag{9}$$

## V. Results

Our model was implemented in Matlab® software using WaveLab 850 toolbox [3], a collection of functions used to implement a variety of algorithms related to wavelet analysis and Neural Network Toolbox, which provides the central tools necessary for the simulation of neural network algorithms.

The data used in this work was obtained by monitoring the incoming and outgoing WiMAX traffic from 67 Base Stations (BS) during 8 weeks, from March 17*th* till May 11*th* 2008[1]. Each BS has its own data set. It consists of numerical values representing the total number of packets corresponding to the uplink traffic recorded every 15 minutes. It results that for a given BS we have the following number of samples: 96 samples/per day, 672 per week, and a total number of 5376 samples for the eight weeks. So, the WiMAX data base can be seen as formed by 67 matrices (one for every BS) that have eight columns (the number of weeks) and 672 lines (the moments of time when the number of packets are recorded in a week).

*One day prediction*

For the purpose of *one day prediction*, we could choose between: the BS number (from 67 possible), the number of samples per day (16, 32 or 96), the number of hours shifted (4, 8, 12 or 24), the day of the week (Monday till Sunday), and between applying or not the GA optimization. We considered 8232 simulations from 11256 possible after combining all of the above possibilities (except of the last one referring to the GA).

Concerning the *Similar Days Selection* model proposed for one day prediction, we used the concatenated raw data representing the samples from a given day of all weeks. If we consider as an example the prediction of the day 3, then we take Wednesdays from the first six weeks. This data was used, after wavelet decomposition, for ANN inputs represented in steps 3 and 4 of the block diagram (figure 2). For ANN target in training process, the data from the Wednesday of the 7*th* week was used. For testing data set, we needed at ANNs' inputs the SWT of the concatenated days from the 2*nd* till the 7*th* week. What we expected from the output of the ANN, was the SWT of Wednesday traffic from 8*th* week.

[1]real Alcatel Lucent network data

TABLE I
Medium RSQ and SMAPE, day prediction

| Model | Hours shifted | Measured Value | 96 samples | 32 samples | 16 samples |
|---|---|---|---|---|---|
| Similar Days | 4 hours | RSQ | 1.212 | 1.249 | 1.317 |
| | | SMAPE | 0.905 | 0.927 | 0.802 |
| | 8 hours | RSQ | **1.159** | **1.120** | **1.205** |
| | | SMAPE | **0.886** | **0.858** | **0.756** |
| | 12 hours | RSQ | 1.330 | 1.219 | 1.287 |
| | | SMAPE | 0.904 | 0.924 | 0.814 |
| | 24 hours | RSQ | 1.178 | 1.201 | 1.263 |
| | | SMAPE | 0.910 | 0.946 | 0.780 |
| All Days | 4 hours | RSQ | 0.715 | 0.739 | 0.711 |
| | | SMAPE | 0.868 | 0.843 | 0.732 |
| | 8 hours | RSQ | **0.820** | **0.782** | **0.792** |
| | | SMAPE | **0.861** | **0.849** | **0.720** |
| | 12 hours | RSQ | 0.634 | 0.767 | 0.690 |
| | | SMAPE | 0.866 | 0.857 | 0.738 |
| | 24 hours | RSQ | 0.741 | 0.781 | 0.703 |
| | | SMAPE | 0.869 | 0.876 | 0.752 |

For the *All Days Selection* model used in one day prediction we take the same example (Wednesday, the last week). In the training phase, we applied at the ANN's inputs the given level of wavelet decomposition of the data that corresponded to Monday, 1*st* week, until Tuesday from 7*th* week. The targeting data consists of the wavelet decomposition obtained from Wednesday, 7*th* week. In the testing phase, we put at ANNs' inputs the decomposition from Monday 2*nd* week, till Tuesday 8*th* week. The obtained output after applying the inverse wavelet transform was compared to the real traffic from the third day of the 8*th* week.

The results are compared in terms of RSQ and SMAPE and are shown in Table I.

We used the GA optimization for day prediction, in order to find the optimal values for the weights between the three layers of the ANNs. The processing time is proportional to the number of links between the layers, therefore this technique was used only in day prediction for the first method. We obtained another 8232 simulations. The average values of this 8232 simulations using the RSQ and SMAPE are presented in Table II.

TABLE II
Medium RSQ and SMAPE, *Similar Days Selection* model, using genetic algorithms

| Hours shifted | Measured Value | 96 samples | 32 samples | 16 samples |
|---|---|---|---|---|
| 4 hours | RSQ | 1.127 | 1.200 | 1.176 |
| | SMAPE | 1.001 | **0.913** | 0.867 |
| 8 hours | RSQ | **1.108** | **1.129** | **1.099** |
| | SMAPE | **0.983** | 0.924 | **0.820** |
| 12 hours | RSQ | 1.211 | 1.155 | 1.168 |
| | SMAPE | 1.008 | 0.951 | 0.851 |
| 24 hours | RSQ | 1.087 | 1.189 | 1.215 |
| | SMAPE | 1.067 | 0.950 | 0.872 |

By comparing the results from these tables, we can see that applying second model for day prediction (all days selection),

TABLE III
MEDIUM RSQ AND SMAPE VALUES BY COMPARING THE DAYS SHIFTED IN ONE WEEK *No Sliding* PREDICTION MODEL

| Samp/Day | Days shifted | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 96 | RSQ | **1.243** | **1.195** | 1.659 | 1.281 | 1.308 | **1.077** | 1.252 |
| | SMAPE | **0.895** | 1.036 | 1.057 | 1.247 | 1.081 | **0.819** | **0.894** |
| 32 | RSQ | **1.318** | **1.276** | 1.415 | 1.392 | 1.401 | **1.106** | 1.380 |
| | SMAPE | 0.992 | 1.0004 | 0.**989** | 1.129 | 1.001 | **0.857** | **0.896** |
| 16 | RSQ | **1.663** | **1.397** | 1.891 | 1.805 | 1.742 | **1.215** | 1.712 |
| | SMAPE | **0.840** | 1.012 | 0.961 | 1.093 | 1.036 | **0.845** | **0.793** |

TABLE IV
COMPARISON BETWEEN THE THREE MODELS BASED ON ANN

| ANN Type | RSQ | SMAPE | MAPE | RMSE | MAE |
|---|---|---|---|---|---|
| No Sliding | 1.137 | **0.946** | 48728 | 1.430 | 0.602 |
| Known Sliding | **0.960** | 1.006 | **25218** | **1.399** | **0.573** |
| UnKnown Sliding | 0.845 | 1.052 | 33783 | 1.738 | 0.648 |

gave us better results with about 2 - 6% in comparison with the first forecasting model (*Similar Days Selection*).

Regarding the use of the genetic optimization for ANN, we can see that the results for SMAPE are worse with about 10% compared to those obtained using ordinary ANN training. But the RSQ values are closer to the ideal 1 when we applied the genetic optimization.

Speaking about the number of samples per day, the best result was obtained when using 16. But in this case, the forecasting technique was not able to predict the sudden increasing and the peaks of our WiMAX traffic. This was because of the way the 16 samples per day were obtained: by making a medium value from other 6 consecutive original samples. Regarding this aspect, 96 samples per day were better to be used instead. In both cases, the optimal number for shifted hours is 8.

*Week prediction*

For *week prediction*, we had 1449 simulations by changing the BS number, the number of samples per day, and the number of shifted days. The results for RSQ and SMAPE are presented in Table III. We present in this table only the results from *No Sliding* model, because we wanted to find out the optimal number of days shifted.

We can see that the best results are obtained using one or six days shifting. Also, the best solutions for forecasting were obtained using 2 or 6 days shifting with 16 samples per day. Anyway, as in the one day prediction model, using 16 samples per day do not give the peaks of the signals.As in the previous cases, the role of SWT is to make the decomposition of data before entering the ANNs. We used the wavelet decomposition from the first seven weeks for ANN training (six for ANN inputs, and one for ANN target).

In testing process, the ANNs inputs contained SWT of the data from the 2*nd* till the 7*th* week, while the traffic from the last week (the 8*th*) is compared to the one obtained after applying the inverse wavelet transform from all the details and approximation taken from ANNs outputs.

## VI. CONCLUSION

In this paper we implemented a technique that combines the wavelet analysis with the ANN model. The main challange was to verify that using the ANNs in WT domain allow us to obtain better results.

For day prediction, we applied genetic optimization techniques to find optimal values for ANN's weights. However, the results in Table I and Table II showed that usual training for ANN has a performance that is greater with about 2-7% than of the genetically optimized ANN. For our database, the best ANN configuration had an overlapping of 2/3 (8 hours shifting), and three neurons for the hidden layer. Between the two models for day forecasting, the *All Days Selection* model was better.

In case of week forecasting we found 12 neurons for the hidden layer to suit our problem. The optimal overlapping was 6/7 and 1/7 (one and six days shifting, see Table III). Between the three forecasting models for week prediction the *Known Sliding* model performed better than the other two as seen in Table IV. To determine this we computed other measures of accuracy, MAPE (Mean Absolute Percentage Error), RMSE (Root Mean Squared Error), and MAE (Mean Absolute Error).

We made comparisons with the results from other articles and found applications that use the same criterias (ANNs with or without wavelets). We didn't take into consideration the type of the predicted data, because in this paper we concentrated on ANNs and their parametrization (the comparison of our models with models that make WiMax traffic predictions without using neural networks is done in [23]). In [27] is given an example of prediction using ANNs without wavelets, but using detrending and deseasonalization preprocessing of the original data instead. The medium results of the best four RMSEs is 2,337, and 1,722 for the best four MAEs, in comparison to 1,399 for RMSE and 0,573 for MAE using our forecasting model. These results prove that applying the proposed ANN configuration with wavelets is a more efficient approach.

The forecasting technique in this paper can be effectively used for building prediction models for time series. However, in order to have higher performance and to reduce the prediction errors, we would recommend to have more data for analysis, to take into consideration the localization of the geographic area where the information has been taken from, and to analyze what types of business is developed in the given region. These aspects represent the start points for our future work in this domain.

**Acknowledgment**
The authors thank Alcatel Lucent, Timisoara, for providing us the WiMAX traffic database and for helpful discussions around WiMAX network. We also thank Prof. Dr. Ing. Alexandru Isar from Politehnica University of Timisoara, Romania and Prof. Dr. Ing. Monica Borda from Tehnical University of Cluj Napoca, Romania for their help and support.

## REFERENCES


[1] *Artificial neural networks, Artificial Intelligence tutorial*, http://www.learnartificialneuralnetworks.com/, october 2009.

[2] *Genetic Algorithm and Direct Search Toolbox*, The MathWorks Inc., 2004-2009.

[3] http://www-stat.stanford.edu/~wavelab/, october 2009.

[4] S. Armstrong, M. Adyaa, *An application of rule based forecasting to a situation lacking domain knowledge*, International Journal of Forecasting, Vol. 16, pp. 477-484, 2000.

[5] N. Clarence, W. Tan, *Incorporating Artificial Neural Network into a Rule-based Financial Trading System*, The First New Zealand International Two Stream Conference on Artificial Neural Networks and Expert Systems (ANNES), University of Otago, Dunedin, New Zealand, November 24-26, 1993.

[6] Daniel S. Clouse, C. Lee Giles, Bill G. Horne, Garrison W. Cottrell, *Time-Delay Neural Networks: Representation and Induction of Finite-State Machines*, IEEE Transaction on Neural Networks, vol. 8, no. 5, September 1997.

[7] M.B.H. Demuth. *Neural Network Toolbox*, The MathWorks Inc., 1992 - 2002.

[8] H. Feng, Y. Shu, *Study on Network Traffic Prediction Techniques*, Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing, Vol. 2, pp. 1041- 1044, 2005.

[9] I. Firoiu, C. Stolojescu, A. Isar, *Forecasting of WiMAX BS Traffic: Observations and Initial Models*, Alcatel Lucent technical repport, January, 2009.

[10] E. Steurer, T.H. Hann. *Much ado about nothing? exchange rate forecasting: Neural networks vs. linear models using monthly and weekly data*, Neurocomputing, vol. 10, pp. 323-339, 1996.

[11] S. Haykin, *Neural Networks: A Comprehensive Foundation*, IEEE Press, McMillan College Publishing Co., 1994.

[12] G. Ibarra-Berastegi, A. Elias, R. Arias, A. Barona, *Artificial Neural Networks vs Linear Regression in a Fluid Mechanics and Chemical Modeling Problem: Elimination of Hydrogen Sulphide in a Lab-Scale Biofilter*, IEEE/ACS International Conference on Computer Systems and Applications, pp.584-587, 2007.

[13] I. Ileana, C. Rotar, A. Incze, *The optimization of feed forward neural networks structure using genetic algorithms*, Proceedings of the International Conference on Theory and Applications of Mathematics and Informatics - ICTAMI 2004, Thessaloniki, Greece, pp. 223-234, 2004.

[14] K. Kannan, D. Venkatesan, *A genetic algorithm-based artificial neural network model for the optimization of machining processes*, Neural Computing and Applications, pp. 135-140, 2008.

[15] B. Krose, P. van der Smagt, *An Introduction to Neural Networks*, The University of Amsterdam, Eighth edition, November 1996.

[16] S. Mallat, *A Wavelet Tour of Signal Processin*, Second Edition, 1999.

[17] A. Mitra, S. Mitra, *Modeling exchange rates using wavelet decomposed genetic neural networks*, Statistical Methodology, vol. 3, issue 2, pp. 103-124, 2006.

[18] Y. Misiti, M. Misiti, *Wavelet Toolbox 4 Users Guide*, MathWorks Inc.,1997 2009.

[19] Nguyen, H. T, Nabney, I. T, *Combining the Wavelet Transform and Forecasting Models to Predict Gas Forward Prices*, ICMLA '08: Proceedings of the 2008 Seventh International Conference on Machine Learning and Applications, IEEE Computer Society, pp. 311-317, 2008.

[20] M.J. Shensa, *Discrete Wavelet Transform.Wedding the a trous and Mallat algorithms*, IEEE Transactions and Signal Processing, 40, pp. 2464-2482, 1992.

[21] L. Smith, P. Tucker, *Distribution of rest days in 12 hour shift systems: impacts on health, wellbeing, and on shift alertness*, Occupational and Environmental Medicine,(56),pp. 206-214, 1999.

[22] C. Stolojescu, A. Cusnir, S. Moga, A. Isar, *Forecasting WiMAX BS Traffic by Statistical Processing in the Wavelet Domain*, Proceedings of the International Symposium on Signals, Circuits and Systems - ISSCS 2009, Iasi, Romania, pp. 177-183, 2009.

[23] C. Stolojescu, S. Moga, P. Lenca, A. Isar, *A wavelet based prediction model for time series*, 4th International Symposium on Communications, Control And Signal Processing, Limassol, Cyprus, 2010, accepted.

[24] T. Taskaya-Temizel, M.C. Casey, *Configuration of Neural Networks for the Analysis of Seasonal Time Series*, ICAPR 2005, LNCS 3686, pp. 297-304, 2005.

[25] T. Taskaya-Temizel, M.C. Casey, *Configuration of Neural Networks for the Analysis of Seasonal Time Series*, Proceedings of the 3rd International Conference on Advances in Pattern Recognition (ICAPR 2005), Lecture Notes in Computer Science 3686, vol. 1, pp. 297-304, 2005.

[26] G. Zhang, B. E. Patuwo, M. Y. Hu, *Forecasting with artificial neural networks: The state of the art*, International Journal of Forecasting 14, pp.35-62, 1998.

[27] G. Peter Zhang, Min Qi, *Neural network forecasting for seasonal and trend time series*, European Journal of Operational Research 160, pp. 501-514, 2005.

[28] Qiang Yang and Xindong Wu, *10 Challenging Problems in Data Mining Research*, International Journal of Information Technology and Decision Making, vol. 5(4), pp. 597-604, 2006.