JOURNAL OF SUSTAINABLE ENERGY VOL. 4, NO. 4, DECEMBER, 2013

A COMPARATIVE STUDY FOR MAXIMUM WIND SPEED PREDICTION

ALBU R.D.*, POPENTIU-VLADICESCU F.**, NAGHIU I.M.***
* University of Oradea, Faculty of Electrical Engineering and Information Technology, Romania
** "UNESCO Chair" Department, University of Oradea, 1 Universităţii Str., Oradea, Romania
*** PhD student, University of Oradea, Oradea, Romania
ralbu@uoradea.ro, popentiu@imm.dtu.dk, iulianaghiu@yahoo.com

Abstract: Wind speed prediction is rapidly becoming one of the most noteworthy parts in wind power integration and operation. In the paper we compare the prediction accuracy of 28 different adaptive models, trained on five wind speed data sets, in order to decide which model is the most appropriate for each data set. We are using real world data collected from five different Romanian cities and presenting detailed experimental results. This research work continues our previous investigations published at ESREL 2013 international conference, where we have proposed a simple and efficient adaptive model for wind speed prediction.

Key words: wind speed forecasting, adaptive models, wind power, prediction

NOTATIONS

LR – Linear Regression
MLP – Multi-layer Perceptron
TLRNN – Time-leg Recurrent Neural Network
TDNN – Time Delay Neural Network
ANM - National Meteorology Administration
DDR3 – Double Data Rate 3
MSE – Mean Square Error
MAE – Mean Absolute Error
RAM – Random Access Memory
FIR - finite impulse response digital
PE – Processing Elements
X = input vector for a layer
W=weight vector
U = input vector for the network, the initial input
n = time index
k = tap index
μ = the feedback parameter
H = Hessian matrix
J = Jacobian matrix
I = identity matrix
c = is always positive, called combination coefficient
e = training error

1. INTRODUCTION

Wind power prediction has a remarkable significance to optimize and increase renewable wind power generation [1]. The most involved problem is to develop more precise prediction models. BP (Backward Propagation) neural networks have been studied comprehensively in order to build wind power prediction models, but results have shown relatively slow convergence rates [2]. Very precise short-term wind speed forecasting is indispensable to reliable power system operations. A very recent study integrates unscented Kalman filter (UKF) with support vector regression (SVR) based state-space model in order to precisely update the short-term estimation of wind speed sequence [3].

2. CONTRIBUTIONS

To improve the reliability of wind power at the small windmill farms and increase predicting accuracy of wind speed, this paper shows a comparative study, implemented and tested on real-world data sets, collected from five different Roumanian cities. The author's contribution of this research work is a study of solutions. For each Romanian city, we investigate which model is the optimal prediction solution, comparing 28 different adaptive models. The 28 models studied here cover almost all classic prediction solutions offered by literature until now. From our knowledge, a comparative study like this was never done before in Romania.

3. METHODOLOGY

To implement, train and test the proposed adaptive models, we made use of Neuro Solutions 6.21 [64 bit] installed on a system with the following parameters: Intel Core i7 2-2.8 GHz processor, 4 GB of DDR3 RAM and 500 GB hard disk. The adaptive models investigated in this study have two inputs, the minimum wind speed and the average wind speed and the desired output used for training is the maximum wind speed. In other words we try to predict the maximum wind speed in the next day based on minimum and average wind speeds recorded in the recent past.

4. EXPERIMENTS AND RESULTS

To train the adaptive models we have collected five data sets from the ANM web site. Each data set contains wind speed values from a different Romanian city. The five cities investigated in this study are: Arad, Cluj, Constanţa, Sulina, and Tulcea [4]. For each of the five cities, the data set consists of three wind speed values: minimum,

ISSN 2067-5534 © 2013 JSE

average and maximum. These values were collected every day, starting with 01.01.2013 to 25.02.2013. Because of limited space reasons, just a sample of the data set in presented in table 1:

Table 1. Sample of data

Data	Minimum wind speed [m/s]	Average wind speed [m/s]	Maximum wind speed [m/s]
01.02.2013	-1	2.5	6
02.02.2013	1	2.7	4
03.02.2013	-1	2.5	7
04.02.2013	-1	1.5	4
05.02.02013	1	1.7	3
…	…	…	…

Each of the 28 adaptive models compared in this study has a name that respect the following syntax:
Topology_name - number_of_hidden_layers - learning_rule - input projection_algorithm
For example, the adaptive model named: MLP-1-LM-SOMIP is a Multi-Layer Perceptron with only one hidden layer, trained using Levenberg-Marquardt learning rule and Self Organizing Map Input Projection algorithm. The first comparison was performed on the data set with wind speed values collected from Arad. The summary of all networks is presented in (fig. 1) and the performance metrics of the best adaptive model are shown in table 2. As we can see, the criteria utilized to determine the best adaptive model are: MSE (Mean Square Error) and MAE (Mean Absolute Error). Since the literature recommends MAE for prediction problems, in our study we have considered MAE as the primary criterion.

As we can see in (fig. 1), Neuro Solutions for Excel selected TLRN-1-O-M (Time-leg Recurrent Neural Network) as the best adaptive model for ARAD data set. The columns from (fig. 1) show that the data set was divided in three parts: training, testing and cross-validation. The training data set is obviously used for training, the cross-validation data set tests the model in the training phase and determines when the training is stopped, while the testing data set is utilized to investigate the prediction accuracy of the model when it receives unseen samples at the input. The most studied TLRN network is the gamma model. The gamma model is characterized by a memory structure that is a cascade of leaky integrators, an extension of the context unit of the Jordan and Elman networks.

Table 2. Performance metrics of TLRN-1-O-M

	Training	Cross Val.	Testing
# of Rows	56	18	10
MSE	0.361936	0.969825	0.531712
Correlation (r)	0.891943	0.652769	0.865755
Min Absolute Error	0.001272	8.46E-06	0.00848
Max Absolute Error	1.702931	3.304567	1.276013
Mean Absolute Error (MAE)	0.469851	0.647543	0.601772

Summary of All Networks

Performance Metrics

Model Name	Training MSE	Training r	Training MAE	Cross Validation MSE	Cross Validation r	Cross Validation MAE	Testing MSE	Testing r	Testing MAE
MLP-1-O-M (Multilayer Perceptron)	0.470855	0.866888	0.563344	0.790675	0.734817	0.628817	1.154233	0.758862	0.738196
LR-0-B-M (Linear Regression)	0.48726	0.846568	0.545623	0.668923	0.779358	0.552318	0.804386	0.822532	0.595468
LR-0-B-L (Linear Regression)	0.427445	0.861259	0.509174	0.735563	0.757267	0.547969	1.152162	0.798859	0.77635
MLP-1-B-L (Multilayer Perceptron)	0.383121	0.879076	0.474042	0.667324	0.780302	0.588588	1.038902	0.776745	0.686109
PNN-0-N-N (Probabilistic Neural Network)	0.339075	0.891886	0.41618	0.784923	0.737131	0.596758	1.264392	0.675264	0.854777
RBF-1-B-L (Radial Basis Function)	0.376277	0.87903	0.479739	0.896721	0.708584	0.642486	1.595972	0.656874	0.950483
GFF-1-B-L (Generalized Feedforward)	0.41141	0.872219	0.492175	0.801698	0.72777	0.649946	0.975585	0.784897	0.671811
MLPPCA-1-B-L (MLP with PCA)	0.739909	0.744962	0.64808	0.566862	0.815689	0.523178	0.960034	0.756784	0.733353
SVM-0-N-N (Classification SVM)	0.812148	0.719712	0.73654	1.506078	0.339511	0.871484	2.444894	0.214201	1.334359
TDNN-1-B-L (Time-Delay Network)	0.839273	0.702819	0.694443	0.882335	0.697637	0.701326	0.64557	0.897669	0.72197
TLRN-1-B-L (Time-Lag Recurrent Network)	0.362248	0.891655	0.483577	0.684316	0.814294	0.692191	1.200417	0.752488	0.784715
RN-1-B-L (Recurrent Network)	0.78356	0.750632	0.646762	1.281884	0.562371	0.704598	1.135174	0.728667	0.843499
MLP-2-B-L (Multilayer Perceptron)	0.330527	0.898483	0.450538	0.823522	0.727228	0.659449	1.318628	0.748375	0.79401
MLP-1-B-M (Multilayer Perceptron)	0.560449	0.820917	0.566461	0.754581	0.744537	0.620067	0.800343	0.822247	0.663178
MLP-2-O-M (Multilayer Perceptron)	0.360845	0.894124	0.463824	0.779533	0.740053	0.637464	1.367921	0.717919	0.759553
MLP-2-B-M (Multilayer Perceptron)	0.521844	0.845317	0.585798	0.829519	0.71264	0.635804	0.989578	0.78835	0.747482
MLPPCA-1-O-M (MLP with PCA)	0.802095	0.740849	0.683852	0.557362	0.825813	0.583498	0.826461	0.782952	0.685649
MLPPCA-1-B-M (MLP with PCA)	1.381842	0.448474	0.886673	1.098838	0.597318	0.813377	0.968152	0.86319	0.815993
GFF-1-O-M (Generalized Feedforward)	0.578159	0.879288	0.624958	0.806808	0.755608	0.665346	1.198381	0.774116	0.805129
GFF-1-B-M (Generalized Feedforward)	0.478852	0.853682	0.534653	0.709861	0.76256	0.592656	0.929786	0.790042	0.69881
RBF-1-O-M (Radial Basis Function)	1.009941	0.654657	0.771552	1.042361	0.644291	0.786517	1.310663	0.724184	0.947695
RBF-1-B-M (Radial Basis Function)	0.511891	0.840308	0.577494	0.98631	0.66322	0.777086	1.497499	0.601926	0.986607
TDNN-1-O-M (Time-Delay Network)	0.354259	0.893985	0.444188	0.987135	0.650357	0.703901	0.792299	0.794582	0.730702
TDNN-1-B-M (Time-Delay Network)	0.301595	0.90828	0.433832	0.83899	0.71609	0.632622	0.944572	0.77019	0.739721
RN-1-O-M (Recurrent Network)	0.664057	0.796076	0.612935	0.888743	0.711748	0.818141	1.275565	0.737698	0.799171
RN-1-B-M (Recurrent Network)	0.768471	0.759981	0.658513	1.434116	0.556421	0.965433	1.458186	0.684572	0.931053
TLRN-1-O-M (Time-Lag Recurrent Network)	0.361936	0.891943	0.469851	0.969825	0.652769	0.647543	0.531712	0.865755	0.601772
TLRN-1-B-M (Time-Lag Recurrent Network)	0.852397	0.721801	0.733078	0.934476	0.668368	0.697153	0.752033	0.848039	0.718948

Fig. 1. Performance of all networks using ARAD data set

The signal at the taps of the gamma memory can be represented by:

$$X_0(n) = U(n) \quad (1)$$

$$X_k(n) = (1-\mu)\, X_k(n-1) + \mu\, X_{k-1}(n-1) \quad (2)$$

The signal at tap k is a smoothed version of the input, which holds the value of a past event, creating a memory. The point in time where the response has a peak is approximately given by k/μ, where μ is the feedback parameter. This means that the neural network can control the depth of the memory by changing the value of the feedback parameter, instead of changing the number of inputs. The parameter μ can be adapted using gradient descent procedures just like the other parameters in the neural network. But since this it is recursive, a more powerful learning rule needs to be applied [5].

For the data set that have wind seed values collected from Cluj, the comparison results are illustrated in (fig. 2). Neuro Solutions for Excel elected as the best model: LR-0-B-M (Linear regression). This is a simple linear regression model that uses backpropagation, have no hidden layers and as gradient search implements Momentum learning rule. The performance metrics for this model are show in table 3.

Summary of All Networks

Performance Metrics

Model Name	Training MSE	Training r	Training MAE	Cross Validation MSE	Cross Validation r	Cross Validation MAE	Testing MSE	Testing r	Testing MAE
MLP-1-O-M (Multilayer Perceptron)	0.291384	0.927473	0.437001	0.913009	0.85757	0.768063	0.335668	0.909992	0.51586
LR-0-B-M (Linear Regression)	0.377658	0.903431	0.471286	0.860436	0.848009	0.725777	0.169143	0.963028	0.34783
LR-0-B-L (Linear Regression)	0.377658	0.903431	0.471286	0.860436	0.848009	0.725777	0.169143	0.963028	0.34783
MLP-1-B-L (Multilayer Perceptron)	0.432524	0.921034	0.567995	0.691481	0.857928	0.689626	0.293285	0.901155	0.517027
PNN-0-N-N (Probabilistic Neural Network)	0.245926	0.938298	0.385995	0.900003	0.825054	0.754557	0.401615	0.934735	0.519524
RBF-1-B-L (Radial Basis Function)	0.279936	0.929414	0.433264	1.020918	0.793964	0.852348	0.384881	0.881373	0.558801
GFF-1-B-L (Generalized Feedforward)	0.865645	0.887964	0.756366	0.373719	0.897878	0.509261	0.336108	0.889965	0.433084
MLPPCA-1-B-L (MLP with PCA)	0.467573	0.906499	0.525446	0.54479	0.847805	0.621045	0.388809	0.89739	0.507324
SVM-0-N-N (Classification SVM)	1.227773	0.683702	0.801686	1.569238	0.428102	0.93703	0.902589	0.781962	0.806058
TDNN-1-B-L (Time-Delay Network)	0.118197	0.975691	0.282742	0.39888	0.935576	0.533662	1.39934	0.500233	0.941544
TLRN-1-B-L (Time-Lag Recurrent Network)	1.512198	0.746517	0.904768	1.513994	0.563109	0.957502	1.659247	0.198837	0.969334
RN-1-B-L (Recurrent Network)	1.226972	0.763223	0.878877	1.448323	0.718377	1.009929	0.918262	0.749768	0.741375
MLP-2-B-L (Multilayer Perceptron)	1.033122	0.81839	0.898695	0.627483	0.82172	0.651073	1.455324	0.147643	1.132686
MLP-1-B-M (Multilayer Perceptron)	0.304159	0.924249	0.429489	0.940362	0.861788	0.773774	0.340007	0.914792	0.516775
MLP-2-O-M (Multilayer Perceptron)	0.281907	0.929604	0.425335	0.994877	0.830921	0.800451	0.474893	0.90557	0.580918
MLP-2-B-M (Multilayer Perceptron)	0.360287	0.912461	0.462752	0.974647	0.82858	0.814656	0.343288	0.89443	0.545962
MLPPCA-1-O-M (MLP with PCA)	1.444068	0.620647	0.830035	2.239497	-0.04343	1.080873	0.867551	0.834851	0.847948
MLPPCA-1-B-M (MLP with PCA)	1.802235	0.423662	0.983731	2.068622	-0.22086	1.071888	0.968832	0.735147	0.904575
GFF-1-O-M (Generalized Feedforward)	0.303255	0.925963	0.449353	0.823748	0.859747	0.732826	0.286527	0.915454	0.491263
GFF-1-B-M (Generalized Feedforward)	0.301337	0.924366	0.431495	1.084256	0.844338	0.813354	0.429533	0.894367	0.571866
RBF-1-O-M (Radial Basis Function)	0.279761	0.930442	0.428006	1.427645	0.639126	0.949923	0.185361	0.932681	0.377536
RBF-1-B-M (Radial Basis Function)	0.721656	0.860931	0.649476	1.069392	0.708081	0.741785	0.831233	0.823836	0.750357
TDNN-1-O-M (Time-Delay Network)	0.267697	0.941022	0.404349	0.654915	0.806813	0.612141	0.349722	0.869425	0.508329
TDNN-1-B-M (Time-Delay Network)	0.226249	0.944535	0.394145	1.300158	0.694606	0.915531	0.507902	0.877993	0.558688
RN-1-O-M (Recurrent Network)	0.288124	0.928407	0.427171	1.037005	0.794907	0.779463	0.30243	0.909014	0.481565
RN-1-B-M (Recurrent Network)	0.314132	0.924132	0.454176	1.081348	0.758565	0.789318	0.280805	0.937965	0.442383
TLRN-1-O-M (Time-Lag Recurrent Network)	0.202407	0.950307	0.369377	1.202375	0.784936	0.821395	0.352245	0.918318	0.481177
TLRN-1-B-M (Time-Lag Recurrent Network)	0.207426	0.948973	0.377346	1.042497	0.803607	0.843072	0.52066	0.869843	0.592044

Fig. 2. Performance of all networks using CLUJ data set

Table 3. Performance metrics of LR-0-B-M

	Training	Cross Val.	Testing
# of Rows	56	18	10
MSE	0.377658	0.860436	0.169143
Correlation (r)	0.903431	0.848009	0.963028
Min Absolute Error	0.017079	0.028946	0.017079
Max Absolute Error	2.235209	2.401678	0.632078
Mean Absolute Error (MAE)	0.471286	0.725777	0.34783

The image below, (fig. 3) shows the comparison results for the data set collected from Constanța. As we can observe, the best network chosen by Neuro Solution for Excel in this case, is LR-0-B-L, an adaptive model very similar to the one chosen for the CLUJ data set. The only difference between the two networks is the learning rule, LR-0-B-L implementing Levenberg-Marquardt instead of Momentum. In order to make sure that the approximated Hessian matrix is invertible, Levenberg–Marquardt algorithm introduces another approximation to Hessian matrix [6, 7]:

$$H \cong J^T J + cI \quad (3)$$

The update rule of Levenberg–Marquardt algorithm can be presented as:

$$W_{k+1} = W_k - (J^T J + cI)\, J_k e_k \quad (4)$$

Summary of All Networks

Performance Metrics

	Training			Cross Validation			Testing		
Model Name	MSE	r	MAE	MSE	r	MAE	MSE	r	MAE
MLP-1-O-M (Multilayer Perceptron)	0.849813	0.88193	0.729688	0.615094	0.750627	0.626672	0.654689	0.82418	0.654312
LR-0-B-M (Linear Regression)	1.227863	0.876395	0.906187	0.429082	0.691412	0.546136	0.872642	0.838779	0.780716
LR-0-B-L (Linear Regression)	0.858896	0.876488	0.722724	0.69224	0.690773	0.616194	0.430363	0.838611	0.46281
MLP-1-B-L (Multilayer Perceptron)	0.93316	0.889778	0.763105	0.521001	0.719307	0.57421	0.914229	0.848886	0.800922
PNN-0-N-N (Probabilistic Neural Network)	0.765156	0.891061	0.657074	0.649537	0.716616	0.647988	0.62018	0.833703	0.628868
RBF-1-B-L (Radial Basis Function)	0.882788	0.876538	0.758156	0.661392	0.767778	0.617578	0.661009	0.835804	0.645819
GFF-1-B-L (Generalized Feedforward)	0.794475	0.890138	0.684161	0.642691	0.701057	0.608676	0.696182	0.846486	0.661308
MLPPCA-1-B-L (MLP with PCA)	2.413661	0.722774	1.294232	0.631881	0.492367	0.551419	2.143274	0.581609	1.229517
SVM-0-N-N (Classification SVM)	2.138482	0.674453	1.09817	0.67882	0.491251	0.69588	2.367443	0.330241	1.146136
TDNN-1-B-L (Time-Delay Network)	1.354606	0.845617	0.890261	0.540114	0.631252	0.628996	0.62311	0.832648	0.646794
TLRN-1-B-L (Time-Lag Recurrent Network)	1.234588	0.874381	0.808714	0.981261	0.547869	0.763863	0.729211	0.746193	0.657437
RN-1-B-L (Recurrent Network)	1.434913	0.808249	0.943337	2.007022	0.555178	1.239804	0.656472	0.859508	0.684596
MLP-2-B-L (Multilayer Perceptron)	1.459139	0.844539	1.018366	0.514524	0.63437	0.510698	1.185196	0.792602	0.926456
MLP-1-B-M (Multilayer Perceptron)	2.13443	0.791567	1.213215	0.585736	0.550554	0.507274	2.012111	0.678528	1.204656
MLP-2-O-M (Multilayer Perceptron)	0.994972	0.869277	0.806859	0.578052	0.709473	0.569841	0.648693	0.825258	0.661272
MLP-2-B-M (Multilayer Perceptron)	0.94392	0.866145	0.764507	0.693666	0.696003	0.634823	0.457523	0.835738	0.536465
MLPPCA-1-O-M (MLP with PCA)	2.00927	0.728988	1.141169	0.955508	0.511342	0.704509	1.935968	0.582075	1.14029
MLPPCA-1-B-M (MLP with PCA)	2.727538	0.559494	1.353876	0.993662	0.261946	0.762025	1.901071	0.335221	1.075747
GFF-1-O-M (Generalized Feedforward)	0.977584	0.876593	0.804173	0.623421	0.720859	0.632465	0.789585	0.816626	0.736978
GFF-1-B-M (Generalized Feedforward)	1.327002	0.866846	0.922614	0.568184	0.681061	0.595403	0.573173	0.819578	0.615394
RBF-1-O-M (Radial Basis Function)	1.975352	0.733566	1.114328	0.995385	0.434544	0.835374	1.085512	0.525727	0.749978
RBF-1-B-M (Radial Basis Function)	2.295388	0.62998	1.198521	1.058997	0.308951	0.642225	0.86236	0.761615	0.759423
TDNN-1-O-M (Time-Delay Network)	0.851696	0.93872	0.717348	1.270191	0.616732	0.943637	1.437119	0.753907	1.011414
TDNN-1-B-M (Time-Delay Network)	0.688549	0.905778	0.681978	0.596109	0.689006	0.627978	0.545089	0.790664	0.615448
RN-1-O-M (Recurrent Network)	0.985474	0.864908	0.794798	0.770113	0.759528	0.760783	0.533886	0.818086	0.554049
RN-1-B-M (Recurrent Network)	2.958613	0.467159	1.41298	0.887143	0.436849	0.76108	1.384669	0.589923	0.923132
TLRN-1-O-M (Time-Lag Recurrent Network)	0.789175	0.893418	0.671985	0.593786	0.677009	0.564703	0.545769	0.796296	0.629051
TLRN-1-B-M (Time-Lag Recurrent Network)	0.841671	0.884151	0.710036	0.438202	0.750506	0.498083	0.523406	0.794944	0.591088

Fig. 3. Performance of all networks using CONSTANȚA data set

Levenberg–Marquardt learning rule is a combination between steepest descent algorithm and the Gauss–Newton procedure. When the combination coefficient is very small the Gauss–Newton algorithm is used while when c is very large the steepest descent method is utilized. Table 4 presets performance metrics of LR-0-B-L.

The data set collected from Sulina offers a new network architecture as the leader of predictors (fig. 4). In the Time Delay Neural Network (TDNN) the memory is a tap delay line, in other words a set of memory locations that store the past of the input. Self-recurrent connections (connections that feed the output of a PE to the input) have also been used as memory, and these are named context units. In Neuro Solutions the time delay line is implemented using TDNNAxon, the fundamental element of all FIR filters.

Table 4. Performance metrics of LR-0-B-L

	Training	Cross Val.	Testing
# of Rows	56	17	11
MSE	0.858896	0.69224	0.430363
Correlation (r)	0.876488	0.690773	0.838611
Min Absolute Error	0.006393	0.016528	0.016528
Max Absolute Error	2.830761	1.722337	1.146318
Mean Absolute Error (MAE)	0.722724	0.616194	0.46281

Performance Metrics

Model Name	Training MSE	Training r	Training MAE	Cross Validation MSE	Cross Validation r	Cross Validation MAE	Testing MSE	Testing r	Testing MAE
MLP-1-O-M (Multilayer Perceptron)	3.000725	0.896302	1.220143	1.912208	0.903033	1.089103	0.396639	0.965885	0.453714
LR-0-B-M (Linear Regression)	5.561779	0.865394	1.525075	2.074973	0.909588	1.111923	0.571259	0.964953	0.63618
LR-0-B-L (Linear Regression)	3.63446	0.869844	1.301186	2.074137	0.926499	1.242181	0.813649	0.943913	0.757934
MLP-1-B-L (Multilayer Perceptron)	2.568982	0.912022	1.222145	2.141731	0.883862	1.128691	0.46608	0.951872	0.51552
PNN-0-N-N (Probabilistic Neural Network)	1.016258	0.965469	0.700628	2.588425	0.860394	1.376233	0.828704	0.96422	0.787575
RBF-1-B-L (Radial Basis Function)	3.13411	0.889349	1.238312	2.611959	0.878955	1.421122	0.517955	0.964997	0.544225
GFF-1-B-L (Generalized Feedforward)	35.10565	-0.2184	3.359408	22.76563	-0.52256	2.975836	3.908858	0.229545	1.284732
MLPPCA-1-B-L (MLP with PCA)	8.016757	0.730952	1.709468	4.064554	0.739087	1.567829	0.745666	0.890409	0.733104
SVM-0-N-N (Classification SVM)	14.47735	0.68747	2.981395	8.41593	0.442274	2.554103	3.489848	0.808393	1.62796
TDNN-1-B-L (Time-Delay Network)	3.186411	0.894055	1.149064	1.90469	0.905003	1.157383	2.0506	0.926993	1.306275
TLRN-1-B-L (Time-Lag Recurrent Network)	2.803624	0.90261	1.298784	3.516492	0.857742	1.595282	1.875013	0.930483	1.234747
RN-1-B-L (Recurrent Network)	25.86836	0.379432	3.702135	20.98889	0.319759	3.301466	21.78104	0.093128	3.215154
MLP-2-B-L (Multilayer Perceptron)	1.860002	0.942517	1.043849	1.783593	0.9642	1.092797	0.80767	0.907092	0.773532
MLP-1-B-M (Multilayer Perceptron)	2.724851	0.90606	1.195574	1.96978	0.910864	1.120974	0.598553	0.966146	0.586501
MLP-2-O-M (Multilayer Perceptron)	2.749905	0.907557	1.237555	2.206815	0.888801	1.165602	0.613984	0.961747	0.59099
MLP-2-B-M (Multilayer Perceptron)	3.54089	0.87665	1.318931	2.369818	0.902089	1.300963	0.98928	0.936363	0.797566
MLPPCA-1-O-M (MLP with PCA)	8.406769	0.699477	1.781177	3.28369	0.7981	1.428685	0.81024	0.879961	0.732531
MLPPCA-1-B-M (MLP with PCA)	11.85563	0.553675	2.104969	4.743858	0.701785	1.876027	1.228874	0.77833	0.89629
GFF-1-O-M (Generalized Feedforward)	3.107995	0.894714	1.25535	1.736447	0.909552	1.016773	0.471589	0.961078	0.563075
GFF-1-B-M (Generalized Feedforward)	3.228564	0.887485	1.284352	2.149598	0.907621	1.20848	0.835015	0.94097	0.741718
RBF-1-O-M (Radial Basis Function)	2.604945	0.912313	1.193721	3.008184	0.837948	1.479224	0.653931	0.92978	0.545841
RBF-1-B-M (Radial Basis Function)	9.253256	0.684214	2.063249	7.361943	0.562924	2.487833	5.873758	0.702582	2.138381
TDNN-1-O-M (Time-Delay Network)	1.310303	0.958354	0.793823	1.525373	0.914065	0.951953	0.319383	0.965135	0.444953
TDNN-1-B-M (Time-Delay Network)	2.951105	0.897669	1.27837	2.435855	0.885287	1.347085	1.100131	0.828614	0.950357
RN-1-O-M (Recurrent Network)	3.927438	0.862874	1.448746	2.748358	0.850627	1.326546	1.482558	0.732139	0.927006
RN-1-B-M (Recurrent Network)	9.407168	0.654876	2.28765	2.15993	0.878757	1.211926	1.387619	0.729473	1.067837
TLRN-1-O-M (Time-Lag Recurrent Network)	2.321407	0.921752	1.135865	2.131802	0.887402	1.318534	1.114821	0.78814	0.953812
TLRN-1-B-M (Time-Lag Recurrent Network)	2.459183	0.915668	1.268825	2.848234	0.838155	1.453468	1.103826	0.808208	0.888058

Fig. 4. Performance of all networks using SULINA data set

The performance metrics of TDNN-1-O-M are shown in table 5. This network had one hidden layer implemented in Neuro Solutions with the TanhAxon that has as activation function the hyperbolic tangent. The learning rule of this adaptive model is Momentum with a Momentum rate of 0.7 and a step size of 0.1 that is bumped with 10%.

Table 5. Performance metrics of TDNN-1-O-M

	Training	Cross Val.	Testing
# of Rows	68	10	8
MSE	1.310303	1.525373	0.319383
Correlation (r)	0.958354	0.914065	0.965135
Min Absolute Error	0.013522	0.14901	0.015912
Max Absolute Error	4.463786	2.938686	0.953851
Mean Absolute Error (MAE)	0.793823	0.951953	0.444953

The last data set investigated in this study was collected from Tulcea. The prediction accuracy of each model is shown in (fig. 5). For this data set, Neuro Solutions for Excel elects as the best predictor the MLP-2-B-M (Multi-layer perceptron). Table 6 illustrates its prediction performance.

Performance Metrics

Model Name	Training MSE	Training r	Training MAE	Cross Validation MSE	Cross Validation r	Cross Validation MAE	Testing MSE	Testing r	Testing MAE
MLP-1-O-M (Multilayer Perceptron)	1.176855	0.865348	0.861871	1.519028	0.901817	1.02717	0.57408	0.826198	0.660319
LR-0-B-M (Linear Regression)	1.086608	0.867751	0.817322	1.534578	0.91046	1.011798	0.535017	0.837283	0.634185
LR-0-B-L (Linear Regression)	0.997553	0.876214	0.808703	1.76394	0.876123	1.131792	0.756978	0.761013	0.722196
MLP-1-B-L (Multilayer Perceptron)	0.911965	0.901857	0.78904	0.978893	0.926689	0.732642	0.977829	0.783854	0.835319
PNN-0-N-N (Probabilistic Neural Network)	0.476203	0.943218	0.540967	1.771978	0.884126	1.167264	1.082972	0.722652	0.804532
RBF-1-B-L (Radial Basis Function)	0.922155	0.894038	0.768402	2.365035	0.811924	1.228931	0.666982	0.807232	0.709196
GFF-1-B-L (Generalized Feedforward)	0.958247	0.898466	0.776763	1.138775	0.909108	0.932376	1.46831	0.734275	1.004249
MLPPCA-1-B-L (MLP with PCA)	2.323332	0.751349	1.270617	3.120054	0.766524	1.479108	1.468864	0.739555	1.077388
SVM-0-N-N (Classification SVM)	2.356222	0.719377	1.190818	5.107775	0.50492	2.069731	1.43256	0.416952	0.941889
TDNN-1-B-L (Time-Delay Network)	1.201699	0.858408	0.688264	3.094965	0.758999	1.426658	2.843435	0.277093	1.531565
TLRN-1-B-L (Time-Lag Recurrent Network)	0.979503	0.882995	0.776036	2.284905	0.83909	1.328697	1.399053	0.531645	0.923877
RN-1-B-L (Recurrent Network)	3.6061	0.544552	1.489809	3.480826	0.692228	1.459911	0.961003	0.722465	0.732285
MLP-2-B-L (Multilayer Perceptron)	0.669221	0.923513	0.672924	2.727871	0.857493	1.265699	0.884373	0.731663	0.741068
MLP-1-B-M (Multilayer Perceptron)	1.361661	0.850162	0.901343	1.175613	0.921897	0.774189	0.65042	0.864545	0.684196
MLP-2-O-M (Multilayer Perceptron)	1.418628	0.868341	0.970813	1.196934	0.918543	0.899468	0.870132	0.863361	0.841757
MLP-2-B-M (Multilayer Perceptron)	1.172599	0.864004	0.855954	1.352827	0.906156	0.928171	0.498589	0.84734	0.634727
MLPPCA-1-O-M (MLP with PCA)	3.552728	0.704682	1.632263	2.354165	0.811437	1.3093	2.197699	0.761428	1.30621
MLPPCA-1-B-M (MLP with PCA)	4.062472	0.371329	1.61995	5.585568	0.461186	2.19133	1.366578	0.597985	0.964125
GFF-1-O-M (Generalized Feedforward)	1.379024	0.871085	0.958955	1.303216	0.908346	0.967334	0.873105	0.840333	0.835771
GFF-1-B-M (Generalized Feedforward)	1.066817	0.874913	0.804054	1.371364	0.910136	0.965386	0.540773	0.831636	0.649422
RBF-1-O-M (Radial Basis Function)	1.027105	0.877975	0.769159	3.943622	0.75262	1.80958	0.780703	0.764803	0.768283
RBF-1-B-M (Radial Basis Function)	2.010856	0.764535	1.073741	3.777971	0.742582	1.560088	0.918701	0.713676	0.767031
TDNN-1-O-M (Time-Delay Network)	1.831142	0.775601	0.978717	1.699147	0.857595	1.036382	1.037897	0.762664	0.699228
TDNN-1-B-M (Time-Delay Network)	1.394211	0.836329	0.916558	1.824044	0.897442	1.238243	0.91479	0.737984	0.671014
RN-1-O-M (Recurrent Network)	1.702169	0.785946	0.960864	1.475733	0.881451	0.99848	1.55375	0.591578	0.914003
RN-1-B-M (Recurrent Network)	1.974542	0.748007	1.073489	1.655618	0.862634	0.947691	1.222681	0.71266	0.82154
TLRN-1-O-M (Time-Lag Recurrent Network)	1.169548	0.860764	0.800959	1.463061	0.883737	0.957165	0.88161	0.768646	0.696668
TLRN-1-B-M (Time-Lag Recurrent Network)	1.015439	0.87942	0.747149	1.337326	0.902997	0.916563	0.787146	0.752993	0.638072

Fig. 5. Performance of all networks using TULCEA data set

Table 6. Performance metrics of MLP-2-B-M

	Training	Cross Val.	Testing
# of Rows	56	11	18
MSE	1.172599	1.352827	0.498589
Correlation (r)	0.864004	0.906156	0.84734
Min Absolute Error	0.021147	0.190774	0.136042
Max Absolute Error	2.78608	2.804189	1.190774
Mean Absolute Error (MAE)	0.855954	0.928171	0.634727

Regularly, for static pattern classification, the MLP with two hidden layers is a widespread classifier. In other words, the discriminant functions can take any shape, as required by the input data clusters. Additionally, when the weights are appropriately normalized and the output values are in [0, 1], the MLP have the maximum performance. In terms of mapping aptitudes, the MLP is supposed to be able to approximate random functions [5].

5. CONCLUSIONS

In this research work we have compared 28 adaptive models in order to find the best wind speed predictor for a particular data set. In our experiments we made use of data sets that have wind speed values collected from five Romanian cities. The results strengthen the theory that the prediction accuracy of each model is strongly depended upon the data set on which it was trained. Linear Regression networks seems to be well-suited for predicting wind speed from Constanța and Cluj cities while recurrent models like Time Delay Networks and Time-leg Recurrent Networks are the optimal solutions for cities like Sulina and Arad. Multi-layer perceptron with two hidden layers appears the top predictor for the Tulcea wind speeds data set. As future research we plan to develop an adaptive models ensemble that integrates the networks, shown in this study as optimal for each data set.

ACKNOWLEDGEMENTS

This work was partially supported by the strategic grant POSDRU/CPP107/DMI1.5/S/80272, Project ID80272 (2010), co-financed by the European Social Fund-Investing in People, within the Sectorial Operational Program Human Resources Development 2007-2013.

REFERENCES

[1] Moyano C, F., Pecas L. J. A., *An optimization approach for wind turbine commitment and dispatch in a wind park.* Electric Power Syst Res, 79(1):71–9, 2009.

[2] Gaofeng F., Weisheng W., Chun L., Huizhu D., *Wind power prediction based on Artificial Neural Network*, Proceedings of the CSEE, vol. 28, no. 34, pp. 118-123, Dec. 2008.

[3] Kuilin Chen, Jie Yu, *Short-term wind speed prediction using an unscented Kalman filter based state-space support vector regression approach*, Applied Energy Elsevier journal, no. 113, pp. 690–705, Available online 31 August 2013.

[4] Albu Răzvan-Daniel, Florin Popențiu-Vlădicescu, *On the Best Learning Algorithm for Web Services Response Time Prediction*, paper accepted at ESREL 2013 "Annual Conference, Advances in Safety, Reliability and Risk Management" , 30 Sept- 2 Oct 2013, AMSTERDAM, Holland.

[5] Neuro Solutions help: http://www.aertia.com/docs/nd/neurosolutionshelp.pdf.

[6] Hao Yu, Bogdan M. Wilamowski, *Levenberg–Marquardt Training*, 14th International Conference on Intelligent Engineering Systems (INES-09), 9/3/2010.

[7] H. Yu and B. M. Wilamowski, *C++ implementation of neural networks trainer*, in 13th International Conference on Intelligent Engineering Systems (INES-09), Barbados, April 16–18, 2009.