Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2017) 25: 4746 – 4756
© TÜBİTAK
doi:10.3906/elk-1704-126

A cooperative neural network approach for enhancing data traffic prediction

Salihu Aish ABDULKARIM[1,*], **Isa Abdullahi LAWAL**[2]
[1]Department of Computer Science, Federal University Dutse, Jigawa, Nigeria
[2]Department of Computer Engineering, Aljouf University, Sakaka, Saudi Arabia

Received: 11.04.2017 • **Accepted/Published Online:** 18.09.2017 • **Final Version:** 03.12.2017

Abstract: This paper addresses the problem of learning a regression model for the prediction of data traffic in a cellular network. We proposed a cooperative learning strategy that involves two Jordan recurrent neural networks (JNNs) trained using the firefly algorithm (FFA) and resilient backpropagation algorithm (Rprop), respectively. While the cooperative capability of the learning process ensures the effectiveness of the regression model, the recurrent nature of the neural networks allows the model to handle temporally evolving data. Experiments were carried out to evaluate the proposed approach using high-speed downlink packet access data demand and throughput measurements collected from different cell sites of a universal mobile telecommunications system-based cellular operator. The proposed model produced significantly superior results compared to the results obtained on the same problems from the traditional method of separately training a JNN with FFA and Rprop.

Key words: Neural networks, cooperative methods, throughput prediction, firefly algorithm, resilient propagation algorithm

1. Introduction

With the recent proliferation of mobile devices with Internet connectivity capabilities and the deployment of mobile data networks globally, the demand for mobile data has increased immensely. According to recent International Telecommunication Union (ITU) statistics [1], the year 2015 witnessed unprecedented growth in information and communication technologies (ICTs) with Internet user penetration increasing by 7-fold since 2000 and 3G mobile broadband penetration globally reaching 47%, a value that showed a 12-fold increase since 2007 [1]. Although the present day mobile networks (e.g., 3G and 4G) can provide high-speed mobile data for users, network operators are still faced with several challenges including how to continually plan and optimize the network resources in order to guarantee quality of service (QoS) and end-user experience. One way to handle this issue is to estimate (predict) the mobile data demand beforehand, so that network resources can be efficiently and adequately distributed to handle the demand.

Mobile data traffic prediction plays an important role in facilitating the optimization and scheduling of mobile data network resources because it helps in estimating the mobile data traffic ahead of time [2]. With a timely estimate of the mobile data demand, network operators, for example, can automatically perform optimization of their antennas in order to adjust the radiation pattern of the antennas to curtail traffic congestion. Long-term (6 months to a year) mobile data traffic prediction is useful for network resource planning and system capacity expansion. Short-term (1 h to weeks) mobile data traffic prediction, on the other hand, is

*Correspondence: sakwami@fud.edu.ng

important for example for short-term network facility maintenance scheduling. This paper is concerned with the latter prediction period. We explore the use of a cooperative neural network approach to improve the prediction of mobile data traffic in a universal mobile telecommunications system (UMTS)-based mobile data network. The paper is organized as follows: Section 2 provides the related work, Section 3 discusses the methodology, and Section 4 presents the proposed prediction model. The experimental procedure followed is described in Section 5. The results are presented and discussed in Section 6. Section 7 concludes the paper.

2. Related work

The problem of data traffic prediction in a cellular network is usually posed as that of regression involving estimating the future data traffic using past measurements or historical data [3]. There are two major approaches for the regression process, namely statistical modeling and machine learning [4,5]. With the statistical modeling approach, the moving averages of past measurements of the data traffic are used to predict future data traffic [6]. For example, Svoboda et al. [7], Yu et al. [8], and Dong et al. [2] studied the performance of autoregressive moving average (ARMA) and dynamic harmonic regression (DHR) for the prediction of data traffic of 3G and 4G mobile networks. Similarly, Yoo and Sim [9] predict the expected bandwidth utilization on high-bandwidth wide area networks using autoregressive integrated moving average (ARIMA). However, ARMA, DHR, and ARIMA rely heavily on user experience and give rise to a regression model that cannot be directly applied to data traffic with unknown characteristics [10]. Machine learning methods such as neural networks (NNs) [11] and support vector regression (SVR) [12], on the other hand, allow the development of a regression model without prior knowledge of the characteristic of the traffic data. For example, Mirza et al. [4] predict the transfer control protocol throughput of a network that comprises multiple paths between data senders and receivers using SVR. Lawal et al. [13] propose a feedforward neural networks (FNNs) ensemble approach for predicting the data traffic in a UMTS-based cellular network. FNN is, however, limited to handling stationary data and since mobile data from the UMTS-based network are often nonstationary due to time varying behavior of the underlying network characteristics its performance can sometimes fall short of the desired accuracy.

Similar to the approach by Lawal et al. [13], in the present paper, we propose a technique that employs NN models cooperating by transferring knowledge and experience during the training process in order to improve the accuracy of the data traffic prediction problem. Unlike Lawal et al. [13], in the present work, we use a recurrent NN specifically designed to handle temporal (i.e. nonstationary) data for reliable mobile data traffic prediction. Moreover, while Lawal et al. [13] employed only one data set, i.e. high-speed downlink packet access (HSPDA) traffic demand, for the evaluation of their proposed approach, we, on the other hand, use the HSDPA traffic demand data set and an additional HSDPA user throughput dataset for evaluation. The user throughput is the speed at which data are transferred at a given period of time and it is used as an important QoS metric in cellular data networks [14].

3. Methodology

NNs are being applied in predicting traffic in high-speed communication networks [15], and, traditionally, the model of choice is a feedforward NN (FNN) trained using a backpropagation algorithm [16]. FNNs are, however, not designed to handle dynamic systems and are therefore limited to handling stationary data. Since practical applications such as data traffic forecasting are often dynamic, recurrent NNs (RNNs), which are specifically designed to handle temporal data, are required for effective modeling. The widely used RNNs in forecasting applications are the Elman NN (ENN) and Jordan NN (JNN), and a recent empirical study has shown the superiority of the JNN over the ENN in forecasting applications [17]. Moreover, the classical backpropagation algorithm is known to be susceptible to premature convergence on a local minimum. Another drawback of backpropagation is that it is heavily dependent on the search starting points. Global training algorithms such as FFA are being used as alternatives, to address the issues of the backpropagation algorithm. The FFA has been shown to be effective in training NN forecasters [18]. The following sections discuss the JNN and the FFA and Rprop algorithms used in this study.

3.1. Jordan neural network

The JNN is a model that realizes functional dependency between sequence elements and estimates on one hand and the to-be forecast value on the other [19]. The JNN is similar to a standard three-layer feedforward NN structure with additional feedback connections from the output layer to a context layer in order to deal with temporal characteristics of sequential data. The context layer serves as an extension to the input layer, and the context node is fully interconnected with all hidden layer nodes. Thus, the input vector,

$$X = \underbrace{x_1, ..., x_{l+1}}_{Input},\ \underbrace{x_{l+2}, ..., x_{l+1+K}}_{Context} \tag{1}$$

The activation function of each output node is calculated as

$$O_k = f_{O_k}\ \left(\sum_{j=1}^{J+1} w_{k,j} f_{y_j}\left(\sum_{i=1}^{I+1+K} v_{j,i} x_i\right)\right) \tag{2}$$

where $(x_{I+2,}, \ldots, x_{I+1+K}) = (O_1\,(t-1)\,, \ldots, O_J(t-1))$.

3.2. Resilient propagation algorithm

The Rprop is an efficient supervised batch learning scheme that performs a direct adaption of weight step based on local gradient information. At the beginning, weights of the NN are initialized randomly to small numbers such that the mean of the weights is approximately zero and all update values are set to initial value δ_0, usually chosen as a small value, e.g., $\delta_0 = 0.1$ [20]. Choice of this value is not critical because it is adapted during the learning process. To prevent overflow/underflow problems, parameters δ_{max} and δ_{min} are respectively used to place restriction on the upper and lower bounds of the update step sizes. At each iteration, if the derivative $\partial E/\partial w_{i.j}$ changes sign, the weight update value, $\delta_{i,j}(t)$, is decreased by the factor η^- (to reduce the effect of the previous large update), while, in contrast, if the derivative retains its sign, the update value is increased by η^+ (to promote convergence) before weight adjustment. Riedmiller [21] suggested $\eta^+ = 1.2$ and $\eta^- = 0.5$ as a good choice. For most problems, Rprop does not require optimizing parameters to obtain optimal or at least near optimal convergence times and this is one of its key strengths [21].

3.3. Firefly algorithm

The FFA is a biologically inspired meta-heuristic global optimization method developed in 2008 by Yang, based on the following assumptions [22]:

- All fireflies get attracted to each other

- Attraction is proportional to brightness, and both decrease as distance increases. For any two fireflies, the less bright one is attracted to the brighter one. Fireflies move randomly when there is no brighter one.
- The brightness of a firefly is determined by the landscape of the objective function optimized.

The variation of attractiveness is defined as [23]

$$\beta(r) = \beta_0 e^{-\gamma r^2} \tag{3}$$

where r is distance and β_0 the attractiveness as $r = 0$, and $\gamma$is light absorption coefficient.

The movement of a firefly p attracted by a brighter one, say q, is determined as

$$x_{pk}(t+1) = x_{pk}(t) + \beta_0 e^{-\gamma r_{pq}^2} \left[x_{qk}(t) - x_{pk}(t)\right] + \alpha(t)\, \epsilon_{pk}(t)\ , \tag{4}$$

where $k = 1 \ldots D$, (D is the dimension of the problem) and $\alpha(t)$ controls the step size, while $\epsilon_p(t)$ is a vector of random numbers at time t. The FF algorithm is summarized in Algorithm 1.

To train a NN with the FFA, each firefly in the swarm represents a weight vector of the NN and the sum square error (SSE) is used to calculate the fitness of each firefly. With this representation and fitness function, the FFA is used to minimize the NN's weight vector.

Algorithm 1 Firefly Algorithm

```
Require: Randomly initialize a population of fireflies
Require: Initialize algorithm parameters a, b0, and g
 while stopping condition(s) are not true do
   for each firefly p = 1 : n do
   for each firefly q = 1 : n do
     if f (xp)  < f (xq) then
       Move firefly p towards q using Eq. 4
       Vary b0 with distance r using Eq. 3
       Evaluate and update solutions
     end if
   end for
  end for
  Rank fireflies and find the current best
end while
```

4. Proposed prediction model

In this paper, a cooperative JNN model is proposed for modeling mobile data traffic. The proposed technique employs two different JNN models cooperating by transferring knowledge and experience during the training process. The structure of the proposed model is shown in Figure 1. A JNN is first trained using the FFA for $I_n/5$ iterations to locate a promising region in the weight space, where I_n is the total number of iterations budgeted for training. The set of weight obtained is then passed to the second JNN for its weight initialization. The second JNN is then trained using Rprop for the remaining $4I_n/5$ iterations to further fine tune the weights. The values $I_n/5$ and $4I_n/5$ for the FFA and Rprop training were respectively found to be optimal via cross validation from the values [1/10, 1/5, 2/5, 3/5, 4/5] considered.

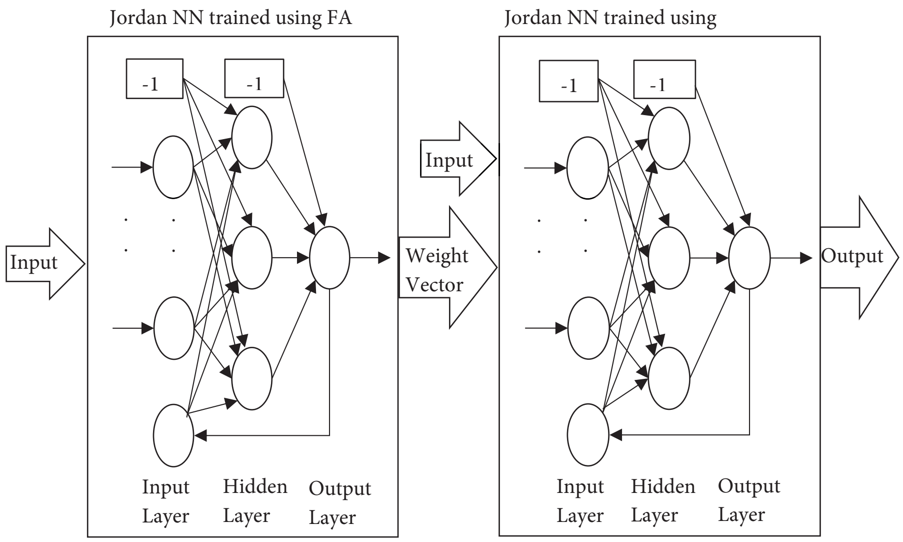


Figure 1. Cooperative firefly-Rprop Jordan neural network.

5. Experimental setup

To evaluate the effectiveness of the proposed model in data traffic forecasting applications, an empirical study comparing it with two models, the JNN trained using Rprop and the JNN trained using FFA, was performed. All experiments were carried out using version 0.9 of the Computational Intelligence Library (CILib). CILib is available at https://github.com/cirg-up/cilib. Section 5.1 describes the dataset used in the study. Section 5.2 lists the control parameters for the algorithms employed. Performance measures and statistical tests used are described in Sections 5.3 and 5.4, respectively.

5.1. Datasets

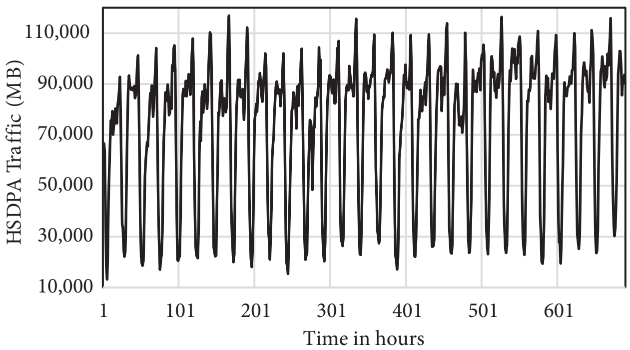
Two datasets were used in the study:

1) HSDPA traffic demand (HSDPA-TD): consists of 690 data points representing the aggregated hourly measurement of HSDPA-TD from 60 different cell sites of a Nigerian UMTS-based cellular operator, from 1 to 30 January 2016. The dataset was first used in [13] and is available at http://tinyurl.com/zcwo7qs. As shown in Figure 2a, the data series is nonstationary, with a slight upward trend, and strong seasonal patterns.

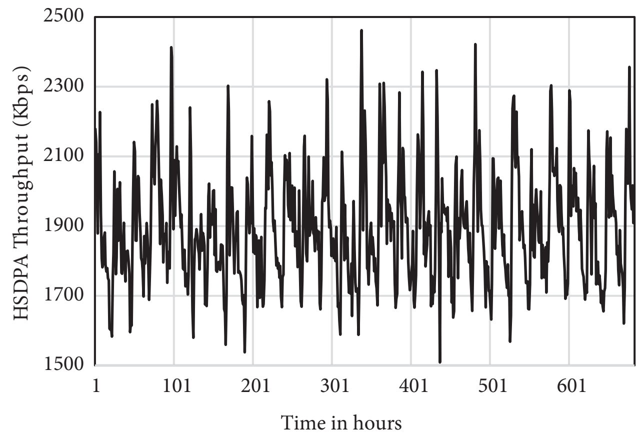
2) HSDPA throughput (HSDPA-T): the HSDPA-T data series has a total of 683 observations representing an aggregate hourly data throughput per user recorded over 36 cell-sites of a Nigerian UMTS-based cellular operator, for a period of 28 days. Illustrated in Figure 2b, the HSDPA-T series has a constant trend and a strong seasonal pattern with no obvious outliers. Another distinct feature of the series is it increases in variability. The HSDPA-T dataset can also be obtained at http://tinyurl.com/zcwo7qs.

To ensure the datasets fall within the active range and domain of the activation functions used, they were scaled to [–1, 1] using linear transformation,

$$x_m = (b - a)\,\frac{x_0 - x_{min}}{x_{max} - x_{min}} + a \tag{5}$$



(a) HSDPA -TD



(b) HSDPA -Th

Figure 2. Samples of the datasets used for the evaluation of the proposed method.

where x_m and x_0 represent the scaled and original data; a and b define the lower and upper limits of the scaled data; x_{min} and x_{max} are the minimum and maximum values of the original data, respectively. For faster convergence as suggested in .....[24], the datasets are further normalized to set the mean of the training data close to zero. Normalization was done using

$$x'_m = \frac{x_m}{\sqrt{M}}, \tag{6}$$

where x'_m is the normalized observation and M is the total number of observations. The first 70% of the dataset was used for training and the remaining 30% for testing.

5.2. Parameters

For each dataset, an NN structure (which involves the number of input, hidden, and output layer nodes) was selected. Other parameters of the training algorithms used were also selected for optimal performance. All these parameters were determined as follows:

1) JNN Architecture; the number of input nodes, n, is fixed at 24, one for each of the 24 hours in a day. This intuitive method was used by several analysts such as [3,25] and has been effective in constructing optimal NN structures. The number of hidden layer nodes, h, was determined using [26]

$$N_h = \frac{(4n^2 + 3)}{(n^2 - 8)} \tag{7}$$

A single output node is used, and the multistep (i.e. the 6-h and 12-h ahead) forecasts considered were generated iteratively using a single step ahead (i.e. one output node) as used in Box–Jenkins model [27]. In the iterative multistep strategy, forecast values are iteratively used as inputs for the next forecasts. In the hidden and output layer nodes, a modified hyperbolic tangent function was employed and is defined as [24]

$$f\left(net\right) = 1.7159 \tanh\left(\frac{2}{3} net\right) \tag{8}$$

This choice was in line with the range the datasets were scaled.

All NNs' weights were randomly initialized in the range $(-\frac{1}{\sqrt{F}} \frac{1}{\sqrt{F}})$, where F is the number of incoming connections for a specific node. Wessels and Barnard [28] have shown this to be a good initialization range.

2) Rprop setup: Default algorithm parameters were adopted because for many problems the parameters lead to optimal convergence [21].

3) FF setup: A swarm of 20 fireflies was used, α was set to vary linearly from 0.2 to 0, β_0 was fixed at 0.2, and γ was set to 1.0. This setting was adopted from [18], where the authors evaluate the effectiveness of the FF algorithm in training a NN forecaster.

5.3. Performance measure

Mean square error (MSE) and mean absolute percentage error (MAPE) are employed to measure the performance of the proposed model. Generalization factor, ρ, is also used to measure the generalization ability of the model. The ρ proposed in [29] indicates the overfitting behavior of a model and it is defined as E_G/E_T, where E_T and E_G are the MSE over the training and generalization sets respectively. Overfitting is a phenomenon where a model performs well on training data but poorly on generalization data. $\rho < 1$ is an indication of good generalization performance while $\rho > 1$ is an indication of overfitting.

5.4. Statistical methods

In the study, 50 independent runs were carried out for each experiment, with the mean E_T and E_G reported. A two-tailed nonparametric Mann–Whitney U test [30] was used to determine whether the difference in performance when comparing results is statistically significant. Tests were evaluated at 95% level of significance. All reported P-values were bounded below by 0.0001 for convenience.

6. Experimental results and discussion

Table 1 summarizes the average E_T and E_G produced by the proposed and benchmark models in 6- and 12-h-ahead prediction of the HSDPA-TD and HSDPA-Th data series, with minimum values displayed in bold. Also reported in the table is the ρ of the forecasting models. Table 2 presents the P-values of the pair-wise comparison of the models investigated. For convenience, the following naming conventions are used in the tables:

- J-Rprop; a model of JNN trained using Rprop
- J-FF; a model of JNN trained using FFA
- J-FFRprop; the proposed model

For the HSDPA-TD and HSDPA-Th problems, the J-FFRprop model produced the lowest average E_T and E_G in both the 6- and 12-h-ahead prediction, significantly outperforming the J-Rprop and J-FF models. The J-FF yielded significantly better results compared to J-Rprop on both E_T and E_G in the 6- and 12-h-ahead prediction of the two data series. All the models studied indicated no sign of overfitting, except for J-Rprop, which showed a slight sign of overfitting in 6-h-ahead prediction of HSDPA-TD, and in 12-h-ahead of HSDPA-Th.

Figure 3 illustrates performance progression of the forecasting models over time, obtained on the HSDPA-TD problem during training and generalization. As visualized in the figures, the J-Rprop model struggles

Table 1. Results of predicting HSDPA-TD and HSDPA-Th data series: means over 30 samples reported.

HSDPA-TD						
Model	6-h-ahead	12-h-ahead				
	E_T	E_G	ρ	E_T	E_G	ρ
J-Rprop	1.32E − 04	1.36E − 04	1.03	1.94E − 04	1.53E − 04	0.79
	(8.10E − 05)	(8.22E − 05)	(0.11)	(1.36E − 04)	(6.25E − 05)	(0.21)
J-FF	2.14E − 05	1.74E − 05	0.81	2.16E − 05	1.79E − 05	0.82
	(1.23E − 06)	(1.18E − 06)	(0.03)	(1.15E − 06)	(9.68E − 07)	(0.05)
J-FFRprop	**1.69E − 05**	**1.48E − 05**	0.88	**1.71E − 05**	**1.50E − 05**	0.88
	(3.11E − 07)	**(2.39E − 07)**	(0.02)	**(3.27E − 07)**	**(3.33E − 07)**	(0.02)
HSDPA-Th						
Model	6-h-ahead	12-h-ahead				
	E_T	E_G	ρ	E_T	E_G	ρ
J-Rprop	1.60E − 04	1.31E − 04	0.82	1.67E − 04	1.84E − 04	1.10
	(1.46E − 05)	(1.35E − 05)	(0.17)	(1.29E − 05)	(4.35E − 05)	(0.21)
J-FF	6.53E − 05	6.14E − 05	0.94	6.94E − 05	6.32E − 05	0.91
	(7.18E − 07)	(9.07E − 07)	(0.01)	(9.08E − 07)	(1.22E − 06)	(0.00)
J-FFRprop	**6.37E − 05**	**5.72E − 05**	0.90	**6.42E − 05**	**5.95E − 05**	0.93
	(7.18E − 08)	**(2.11E − 07)**	(0.01)	**(6.11E − 08)**	**(3.41E − 07)**	(0.02)

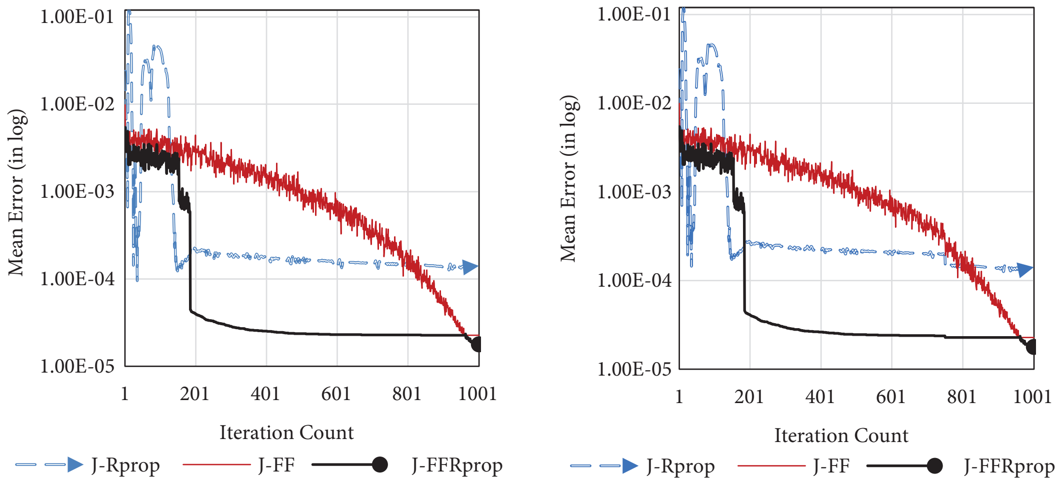
Table 2. Mann–Whitney P-values obtained for the average ET and E G comparisons with reference to the null hypothesis that the means of the compared samples are equal at the significance level of 95%.

Problem	Model	P-values			
		6-h-ahead		12-h-ahead	
		E_T	E_G	E_T	E_G
HSDPA-TA	J-Rprop vs. J-FF	0.0001	0.0001	0.0001	0.0001
	J-Rprop vs. J-FFRprop	0.0001	0.0001	0.0001	0.0001
	J-FF vs. J-FFrprop	0.0001	0.0001	0.0001	0.0001
HSDPA-Th	J-Rprop vs. J-FF	0.0001	0.0001	0.0001	0.0001
	J-Rprop vs. J-FFRprop	0.0001	0.0001	0.0001	0.0001
	J-FF vs. J-FFrprop	0.0001	0.0001	0.0001	0.0001

for about 200 iterations to locate a local minimum, due to the problem of search starting point peculiar to backpropagation methods. However, the algorithm efficiently followed the error gradient to find the minimum.

For the J-FF model, the error keeps reducing gradually with iterations, which indicates the training algorithm maintains a good continuous exploration/exploitation trade off in the search space. The step size α_t of the FF algorithm was set to vary linearly from 0.2 to 0, causing the fireflies to have initial large steps (promoting exploration) that linearly decrease to a small value (facilitating exploitation). As evident in the figure, the proposed model exploits the strength of both the Rprop and FF training methods, to emerge as the best model. The J-FFRprop model uses the initial exploration capacity of the FF algorithm to locate a promising region of the search space and then use Rprop's ability to fine tune the weights in that region.

Table 3 compares the actual values of the HSDPA-TD and HSDPA-Th datasets with their forecasted values obtained from the proposed model and the benchmarks. This is further illustrated in a forecast diagram presented in Figure 4. The J-FFRprop model yielded the lowest MAPE and MSE in predicting both the HSDPA-TD and HSDPA-Th.



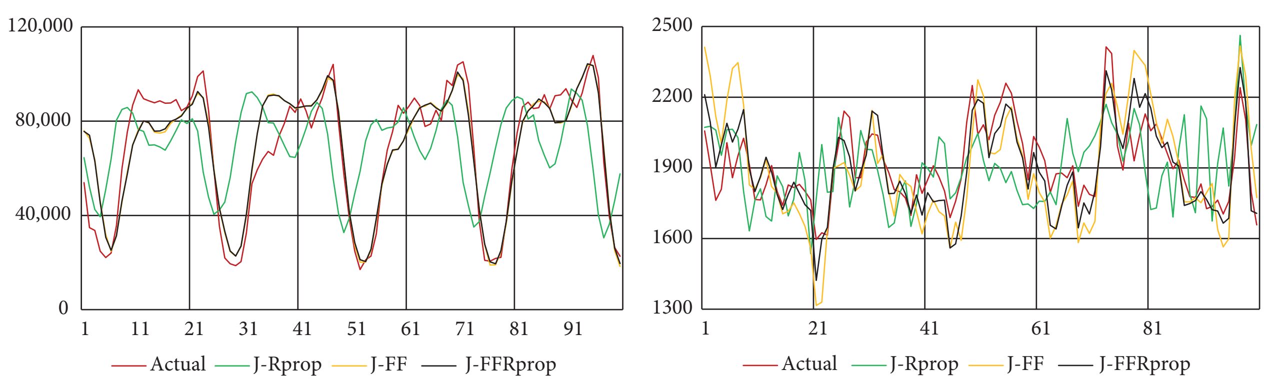
(a) HSDPA - TD: Training error

(b) HSDPA - Th: Generalization error

Figure 3. Training and generalization error results for HSDPA-TD 6-horizon.

Table 3. Results of predicting HSDPA-TD and HSDPA-Th data series on original scale.

Problem	Model	6-h-ahead		12-h-ahead	
		MAPE	MSE	MAPE	MSE
HSDPA-TD	J-Rprop	9.85	6.24E + 07	9.95	8.70E + 08
	J-FF	9.74	6.43E + 07	9.64	6.33E + 07
	J-FFRprop	9.56	6.22E + 07	9.52	6.14E + 07
HSDPA-Th	J-Rprop	8.58	4.19E + 04	5.50	1.85E + 04
	J-FF	8.49	4.10E + 04	4.75	1.47E + 04
	J-FFRprop	8.32	3.63E + 04	4.41	1.29E + 04



(a) Forecast diagram for HSDPA -TD 12-horizon

(b) Forecast diagram for HSDPA -Th 6-horizon

Figure 4. Forecast diagram for HSDPA-TD 6-horizon and HSDPA-Th 12-horizon.

7. Conclusion

This paper proposed a new prediction model based on a cooperative NN strategy in order to improve the prediction accuracy of mobile data prediction in UMTS-based mobile data networks. The model employs two different JNN models cooperating by transferring knowledge and experience during the training process.

A JNN is first trained using the FF algorithm for a number of iterations to locate a promising region in the weight space. The set of weights obtained is then passed to the second JNN for its weight initialization. The second JNN is then trained using Rprop to further fine tune the weights. Experiments were carried to evaluate the performance of the proposed model (i.e. J-FFRprop) in the prediction of two different HSDPA data series. The results obtained were compared with the results of two monolithic models, J-Rprop and J-FF. The J-FFRprop produced a significantly better result in 6- and 12-h-ahead prediction of the data series used. The proposed model also shows no sign of overfitting.

Acknowledgments

This work is completed in part with the computing resources provided by Federal University Dutse, Nigeria, and Al Jouf University, Sakaka, Saudi Arabia

References

[1] International Telecommunication Union. ICT facts and figures - The world in 2015.

[2] Dong X, Fan W, Gu J. Predicting LTE throughput using traffic time series. ZTE Communications 2015; 4: 014.

[3] Liu Y, Lee JY. An empirical study of throughput prediction in mobile data networks. In: IEEE 2015 Global Communications Conference; 6 Dec 2015; IEEE. pp. 1-6.

[4] Mirza M, Sommers J, Barford P, Zhu X. A machine learning approach to TCP throughput prediction. IEEE ACM T Network 2010; 18: 1026-1039.

[5] Rattaro C, Belzarena P. Throughput prediction in wireless networks using statistical learning. In: Latin-American Workshop on Dynamic Networks; 4 Nov 2010; 4p.

[6] He Q, Dovrolis C, Ammar M. On the predictability of large transfer TCP throughput. Comput Netw 2007; 51: 3959-3977.

[7] Svoboda P, Buerger M, Rupp M. Forecasting of traffic load in a live 3G packet switched core network. In: IEEE 2008 International Symposium on Communication Systems, Networks and Digital Signal Processing; 25 Jul 2008; IEEE. pp. 433-437.

[8] Yu Y, Song M, Fu Y, Song J. Traffic prediction in 3G mobile networks based on multifractal exploration. Tsinghua Sci Technol 2013; 18: 398-405.

[9] Yoo W, Sim A. Time series forecast modeling on high-bandwidth network measurements. J Grid Comput 2016; 14: 463-476.

[10] Benet CH, Kassler A, Zola E. Predicting expected TCP throughput using genetic algorithm. Comput Netw 2016; 108: 307-322.

[11] Demuth HB, Beale MH, De Jess O, Hagan MT. Neural Network Design. 2nd ed. USA: Martin Hagan, 2014.

[12] Bao Y, Xiong T, Hu Z. Multi-step-ahead time series prediction using multiple-output support vector regression. Neurocomputing 2014; 129: 482-493.

[13] Lawal IA, Abdulkarim SA, Hassan MK, Sadiq JM. Improving HSDPA traffic forecasting using ensemble of neural networks. In: IEEE 15th IEEE International Conference on Machine Learning and Applications; 18 Dec 2016; IEEE. pp. 308-313.

[14] Blaszczyszyn B, Jovanovicy M, Karray MK. How user throughput depends on the traffic demand in large cellular networks. In: IEEE 12th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks; 12 May 2014; IEEE. pp. 611-619.

[15] Park YK, Lee C. Applications of neural networks in high-speed communication networks. IEEE Commun Mag 1995; 33: 68-74.

[16] Chabaa S, Zeroual A, Antari J. Identification and prediction of internet traffic using artificial neural networks. J Intell Learn Syst Applic 2010; 2: 147.

[17] Abdulkarim SA. Time series prediction with simple recurrent neural networks. Bayero J Pure Appl Sci 2016; 9: 19-24.

[18] Abdulkarim SA, Garko AB. Effectiveness of firefly algorithm based neural network in time series forecasting. Bayero J Pure Appl Sci 2016; 9: 6-10.

[19] Dorffner G. Neural networks for time series processing. Neural Netw World 1996; 6: 447-468.

[20] Riedmiller M, Braun H. A direct adaptive method for faster backpropagation learning: the RPROP algorithm. In: IEEE International Conference on Neural Networks, 1993; IEEE. pp. 586-591.

[21] Riedmiller M. Rprop-Description and implementation details: technical report. Inst. f. Logik, Komplexität u. Deduktionssysteme; 1994.

[22] Yang XS. Firefly algorithm, stochastic test functions and design optimisation. Int J Bio-Inspir Com 2010; 2: 78-84.

[23] Yang XS. Firefly algorithms for multimodal optimization. In: International Symposium on Stochastic Algorithms; 26 Oct 2009; Springer. pp. 169-178.

[24] LeCun Y, Bottou L, Orr GB, Müller KR. Efficient backprop. In: Orr GB, Müller KR, editors. Neural Networks: Tricks of the Trade. Berlin, Germany: Springer, 2012, pp. 9-48.

[25] Hamzaçebi, C. Improving artificial neural networks' performance in seasonal time series forecasting. Inform Sciences 2008; 178: 4550-4559.

[26] Sheela KG, Deepa SN. Review on methods to fix number of hidden neurons in neural networks. Math Probl Eng 2013; 2013, Article ID 425740, 11 pages. doi:10.1155/2013/425740.

[27] Pankratz A. Forecasting with Univariate Box-Jenkins Models: Concepts and Cases. New York, NY, USA: Wiley, 2009.

[28] Wessels LF, Barnard E. Avoiding false local minima by proper initialization of connections. IEEE T Neural Networ 1992; 3: 899-905.

[29] Röbel A. The Dynamic pattern selection algorithm: Effective training and controlled generalization of Backpropagation neural networks: tech rep. Technische Universität Berlin, 1994.

[30] Mann HB, Whitney DR. On a test of whether one of two random variables is stochastically larger than the other. Ann Math Stat 1947, 18: 50-60.