Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 170 (2020) 362–369

Procedia Computer Science

www.elsevier.com/locate/procedia

The 11th International Conference on Ambient Systems, Networks and Technologies (ANT)
April 6 - 9, 2020, Warsaw, Poland

A New Workload Prediction Model Using Extreme Learning Machine and Enhanced Tug of War optimization

Thieu Nguyen[a], Bao Hoang[a], Giang Nguyen[b], Binh Minh Nguyen[a,*]

[a]*School of Information and Communication Technology, Hanoi University of Science and Technology, Hanoi, Vietnam*
[b]*Institute of Informatics, Slovak Academy of Sciences, Bratislava, Slovakia*

Abstract

Time series data is widely accessible in many life areas like economy, weather, stock price, retail sales, distributed system workloads. While many studies have focused on improving existing prediction techniques on accuracy aspect, less efforts is paid towards simple but efficient forecasting models in order to keep the balance between computation cost and prediction accuracy. In this work, we propose a novel time series prediction model, which aims to both model simplicity and accuracy. The core of the model is built based on extreme learning machine. Due to the random determination process for input weights and hidden biases, extreme learning machine requires a large number of hidden neurons to achieve good results and this increases the model complexity. To overcome this drawback, we propose a new opposition-based tug of war optimization to select optimally input weights, and hidden biases then apply to extreme learning machine. Two real public traffic monitoring datasets from Internet service providers were employed to evaluate our design. The achieved outcomes demonstrate that our proposed solution works effectively with satisfied performance in comparison with existing models for distributed system workload prediction.

Keywords: meta-heuristics; opposition-based tug of war optimization; extreme learning machine; internet traffic; workload prediction

1. Introduction

Time series data is a sequence of related data points gathered over time. The goal of time series data prediction process is to make a forecast of data points in the near future based on the historical data. This process is useful for many commercial and industrial applications such as economy forecast, sale improvement, budgetary analysis, stock trading, yield projections, process and quality control of systems, inventory studies, workload projections, census analysis and many more. Because there are many methods, which are used to model and predict time series data, two criteria that are often taken into consideration when applying a forecast technique, including accuracy and effective-

* Corresponding author. Tel.: +84-24-3869-6124 ; fax: +84-24-3869-2906
E-mail address: minhnb@soict.hust.edu.vn

1877-0509
10.1016/j.procs.2020.03.063

ness [1, 23]. While the precision can be measured by many standard metrics according to each prediction method, the effectiveness is difficult to quantify by parameters. For instance, in addition to the accuracy, simpleness, learning speed, the computation cost are also important factors that affects the effectiveness of time series prediction models.

In this paper, we focus on developing a novel time series prediction technique, which can tackle the non-linear models and achieve the same level or even better accuracy, stability, time, and interpretability as compared with other modern methods. The prediction model thus can be used to many applications, especially in this work we use for forecasting workloads of distributed systems. In this direction, we contribute a new version for extreme learning machine (ELM) using our proposed opposition-based tug of war optimization (OTWO) algorithm to select input weights, and using MoorePenrose generalized inverse to calculate output weights. The new gained prediction technique is called OTWO-ELM. The reasons for ELM choice is that this model has several advantages such as simple, interpretable, fast learning, and can bring good quality forecast results. In our model, the improvement of OTWO mainly focuses on decreasing the norm of output weights of ELM and constraining input weights and hidden biases within a reasonable range to enhance the convergence performance of that neuron network. Hence, it improves the prediction accuracy for our model. OTWO-ELM was experimented and evaluated with two real datasets covering the Internet traffic data from European Union (EU) and United Kingdom (UK) cities provided by Internet service providers. We also compared the obtained performance of OTWO-ELM with several other well-known prediction techniques. The gained results showed that our proposals bring positive effectiveness in predicting workloads for distributed systems.

The structure of this paper is as follows. In Section 2, we classify and analyze existing studies to highlight our contributions. Section 3 describes OTWO-ELM design. In Section 4, we present experiments and evaluations for the proposed model to prove its effectiveness. Section 5 concludes and defines our future work directions.

2. Related Work

Artificial neural networks are an effective universal approximator widely used for machine learning regression and classification [29]. However, artificial neural networks often use gradient-based learning algorithms like back-propagation, which causes many drawbacks [2], for example, easily trapped in local minimum, time-consuming computation due to improper learning steps. In addition, the performance of artificial neural networks often is not very stable [32]. Meanwhile, deep neural networks have more hidden layers and bring good performance [17]. The most well-known deep neural networks type for time series forecasting is recurrent neural network [7] as well as its variants like long short term memory [14], and gated recurrent unit [3]. However, there are still two main concerns toward deep neural network usage: 1) requirement of a larger amount of data to train [33] in order to gain more accuracy and 2) the computational cost due to the model complexity [26].

In order to overcome drawbacks and concerns, we consider ELM [10] in this work. It randomly chooses the input weights and hidden biases, determines the output weights of single-layer feed-forward networks through the simple generalized inverse operation of the output matrix of hidden layer. ELM not only learns much faster with better generalization performance but also avoids many difficulties faced by gradient-based learning methods such as stopping criteria, learning rate, learning epochs, and local minimum. The interpretable of ELM has been proved in term of both universal approximation and classification capabilities in [8, 9]. However, ELM still tends to require more hidden neurons than traditional gradient-based learning algorithms. This also leads to ill-condition problem due to randomly selecting input weights and hidden biases. In [34], the authors proposed an ELM version using the differential evolutionary algorithm to select input weights, and using MoorePenrose generalized inverse to analytically determine output weights. These improvements can bring good performance and make a compact ELM network. In this direction, several meta-heuristic algorithms have been proposed to combine with ELM such as particle swarm optimization (PSO) [30], genetic algorithm (GA) [31] to improve ELM performance. However, PSO weaknesses are pointed out [16, 25] as easily to be trapped in the local optima when solving complex multi-modal problems. For GA, many studies such as [20, 24] identified its issues. Concretely, with the high epistatic targeting functions, the technique performance degradation is often large. It is suitable to mention hyper-heuristic strategy for hybridization of nature inspired algorithms like [6], which is designed to efficiently solve computational search problems.

Recently, a novel meta-heuristic called tug-of-war optimization (TWO) based on physics laws in the tug of war game in competitions was proposed in [11]. The authors have proved their effectiveness in several aspect such as solving the structural damage identification problems [13], applying to the optimal design of castellated beams [12].

Even with high performance in solving some problems, like other meta-heuristic algorithms, the traditional TWO can get stuck in local optimal point, and therefore the convergence becomes slow and time-consuming. These problems occur because some solutions update their positions based on the best solution. However, the optimal solution might be in the opposite direction to the current solution. Considering such drawbacks, we propose an improvement of TWO by searching global solution in both directions (forward and opposite). This proposal thus is called opposition-based tug of war optimization (OTWO for short). Based on our knowledge, there is no research to enhance TWO by using opposition-based learning, and no study improves the ability of ELM by OTWO. Our main contributions include:

1. Proposing an improvement of tug of war optimization based on opposition-based learning called OTWO;
2. Proposing an improvement of ELM which using opposition-based tug of war optimization to find good input weights and hidden biases (called OTWO-ELM);
3. Experimenting and evaluating several models including well-known multilayer neural network (MLNN), traditional ELM, GA-ELM, PSO-ELM, TWO-ELM and OTWO-ELM for the system workload prediction problem (in this study, Internet traffic is used as test datasets).

3. Designing Opposition-based Tug of War Optimization Extreme Learning Machine

3.1. Tug of War Optimization

Tug of War Optimization (TWO) is a multi-agent meta-heuristic technique, which simulates each solution $X_i = x_{i,j}$ (Eq. 1) as a team, which participates in a series of tug of war matches. The team's weight is defined based on the quality of that corresponding solutions (Eq. 1), and the amount of pulling force, which a team can deploy on the rope, which is assumed to be proportional to its weight (Eq. 2). Naturally, a lighter weight team will be pulled towards a stronger weight team. The principle helps form the converging operator based on Newtonian (mechanics) laws in TWO algorithm (Eq. 2 to 4). The convergence operator will improve the quality of each solution based on maintaining a reasonable balance between exploration/exploitation phase. The details of TWO is presented in [11].

$$x_{ij}^0 = x_{j,min} + rand(x_{j,max} - x_{j,min}) \qquad j = 1, 2, ..., n \qquad W_i = 0.9\left(\frac{fit_i - fit_{worst}}{fit_{best} - fit_{worst}}\right) + 0.1 \qquad i = 1, 2, .., N \tag{1}$$

$$F_{p,ij}^k = max\{W_i\mu_s, W_j\mu_s\} \qquad F_{r,ij}^k = F_{p,ij}^k - W_i^k\mu_k \tag{2}$$

$$g_{ij}^k = X_j^k - X_i^k \qquad a_{ij}^k = \frac{F_{r,ij}^k}{W_i^k\mu_k}g_{ij}^k \tag{3}$$

$$\Delta X_{ij}^k = \frac{1}{2}a_{ij}^k\Delta t^2 + \alpha^k\beta(X_{max} - X_{min}) \odot randn(1, n) \qquad \Delta X_i^k = \sum_{j=1}^{N}\Delta X_{ij}^k \tag{4}$$

$$X_i^{k+1} = X_i^k + \Delta X_i^k = X_i^k + \sum_{j=1}^{N}\Delta X_{ij}^k \qquad x_{ij}^{k+1} = GB_j + \frac{1}{k}randn(GB_j - x_{ij}^k) \tag{5}$$

where x_{ij}^0 is the initial values of j^{th} variable of the i^{th} candidate solution; $x_{j,max}, x_{j,min}$ are the maximum and minimum values of j^{th} variable respectively; *rand* is random number from a uniform distribution in the interval [0, 1]; n is the number of optimization variables; fit_i is the fitness value of i^{th} solution; fit_{worst}, fit_{best} are the worst and the best fitness values of teams; μ_s, μ_k are the static coefficient and kinematic coefficient of friction; $W\mu_s$ is the pulling force is assumed equal to its static friction force; $F_{p,ij}^k$ is the pulling force between $team_i$ and $team_j$ (Newton's third law); $F_{r,ij}^k$ is the resultant force of heavier $team_j$ on $team_i$ in k^{th} iteration; g_{ij}^k is the gravitational acceleration constant in Newton's second law; X_i^k, X_j^k are the position vector of solution i and j in k iteration; ΔX_{ij}^k is the displacement of $team_i$ after competing with $team_j$;α^k is to gradually decrease the random portion of the team's movement; $\alpha^k \in [0.9, 0.99]$; β is a scaling factor which can be chosen from the interval $(0, 1]$. $\odot$ is the element wise element multiplication; X_i^{k+1} is the new position of $team_i$ after a complete round; GB_j is the best solution so far; *randn* is a random number drawn form a standard normal distribution.

3.2. Opposition-based Tug of War Optimization

Opposition-based Learning (OBL) is used to improve the convergence of meta-heuristic methods to find the global solution of the optimization problem [28]. In general, OBL indicates that for finding the unknown optimal solution, searching both a random direction and its opposite simultaneously gives a higher chance to find the promising regions and to enhance the algorithm performance [15]. In the way, our improvement for TWO covers two stages above. Firstly, the OBL is employed in the initialization of teams to enhance the rate of convergence and avoid stuck in local optima through searching solutions in the whole domain. Secondly, in the updating stage of the population solution, the OBL is used to check if the update in the opposite direction is better than the current ones. These two stages are explained in detail as follows.

Initialization Stage: In general, meta-heuristic algorithms usually begin with initializing a random population, which is an important starting point. If the diversity of the initial population is poor or all solutions are in the local optima region, the algorithms may be led to early convergence and converge at local optima. To ensure the appropriate diverse for the population, we propose the following schema: 1) Randomizing teams (population) S of size $N/2$ based on Eq. 1. 2) Generating opposite teams $\overline{S}$ using OBL to create the opposite solution in S using Eq. 6. 3) With two populations S and $\overline{S}$, we have N randomly solutions.

Update Stage: In this stage, once the teams of a league compete against each other for a complete round, the league should be updated. This is done by comparing the new candidate solutions (the new positions of the teams) to the current teams of the league. If the new candidate solution i is better than the old one in terms of fitness value, the old one is replaced by the new solution. If not, we generate the opposite solution of the old solution using Eq. 7, then comparing the fitness between the old solution and the opposite of the old solution. After this stage, which solution has better fitness value will be kept.

$$\overline{x}_{ij} = u_j + l_j - x_{ij}, \qquad i = 1, 2, .., N \quad and \quad j = 1, 2, ..., n \tag{6}$$

$$\overline{x}_{ij} = u_j + l_j - GB_{ij} + rand(GB_{ij} - x_{ij}) \tag{7}$$

where x_{ij} is the i^{th} point of the j^{th} solution of S; $\overline{x}_{ij}$ is the opposite solution of the x_{ij} solution of $\overline{S}$; u_j and l_j are the upper and lower bound of the j^{th} variable, respectively; GB_{ij} is the global best of the current iteration.

The final development of OTWO is illustrated in the Algorithm 1.

3.3. Opposition-based Tug of War Optimization Extreme Learning Machine (OTWO-ELM)

Our proposed prediction model includes two main algorithms, which are ELM and OTWO as presented in Fig. 1. As mentioned in related work section, ELM was proposed to overcome the drawbacks of gradient-based methods in single-hidden layer feed-forward neural networks by implementing two phases. 1) Phase 1: calculating weights between input and hidden layer by choosing randomly; 2) Phase 2: calculating weights between hidden and output layer by simple generalized inverse operation of the hidden layer output matrix.

The strength of ELM is speed because it requires little time to learn the relation between input and output by random process. However, this is a trade-off between the speed and generalization performance. Non-optimal input weights may be randomly chosen, and this causes bad performance. In order to tackle the problem, in this paper, OTWO is used to replace the random process in the Phase 1 (Fig. 1). There are two important operations in Fig. 1, which are encoding and decoding. The encoding operation encodes weights and biases of the input and hidden layer into a solution/team (real-value vector). Otherwise, the decoding operation decodes the solution into weights and biases of input and hidden layer. In order to avoid long training time problem, a termination criterion with completing a maximum epochs number (training cycles) or achieving error goal is designed. Operations of OTWO in OTWO-ELM model are introduced in Algorithm 1.

4. Experiment and Evaluation

There are two aspects that we consider in our experiments to evaluate the proposals, namely prediction accuracy and stability of models. In this way, tested prediction models consist of well-known multi-layer neuron network (MLNN), traditional ELM, several variants of ELM cover GA-ELM, PSO-ELM, TWO-ELM, and our proposed OTWO-ELM. The related materials and the implementation of our experiment can be found in [19, 18].

Algorithm 1 Opposition-based Tug of War Optimization

Input: The population size p_s , the maximum number of generations g_{max}
Output: The best team

Initialize population following the strategy "Initialization stage" in 3.2
Calculate the weights W of the teams based on Eq. 1 (part 2)
Find the global best GB solution
g ← 1
while $g \leq g_{max}$ **do**
    for each $team_i$ **do**
        for each $team_j$ **do**
            if $W_i < W_j$ **then**
                Move $team_i$ towards $team_j$ using Eq. 4 (part 2)
        Find the total displacement of $team_i$ based on Eq. 4
        Determine the final position of $team_i$ using Eq. 5 (part 1)
        Amend solution when its outside of the search space using Eq. 5
    for each $team_i$ **do**
        if $team_i^{new}$ has better fitness than $team_i^{old}$ **then**
            Replace $team_i^{old}$ by $team_i^{new}$
        else
            Find the opposite position of $team_i^{old}$ based on Eq. 7
            Keep $team_i^{old}$ or replace by the opposite one based on fitness
    Update team weights
    Find the current best solution
    Keep the global best or replace by the current best solution based on fitness
Return GB

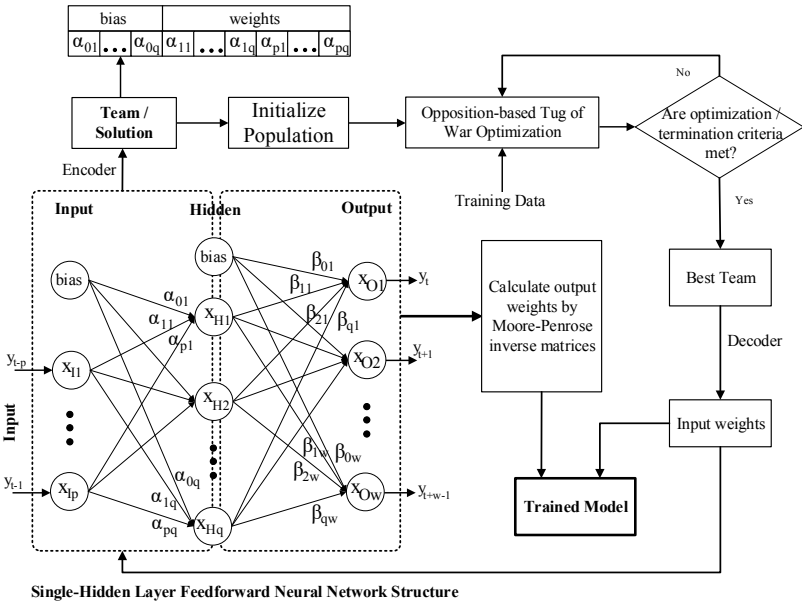

Fig. 1: OTWO-ELM training process

Datasets: In this work, we used two datasets [5]. The Internet traffic (in Megabyte) from data center in European (EU) cities and Internet traffic (in Bytes) from United Kingdom (UK) academic network backbone. Both datasets were collected at five-minute intervals. Both datasets are preprocessed into the form of time-series through normalization

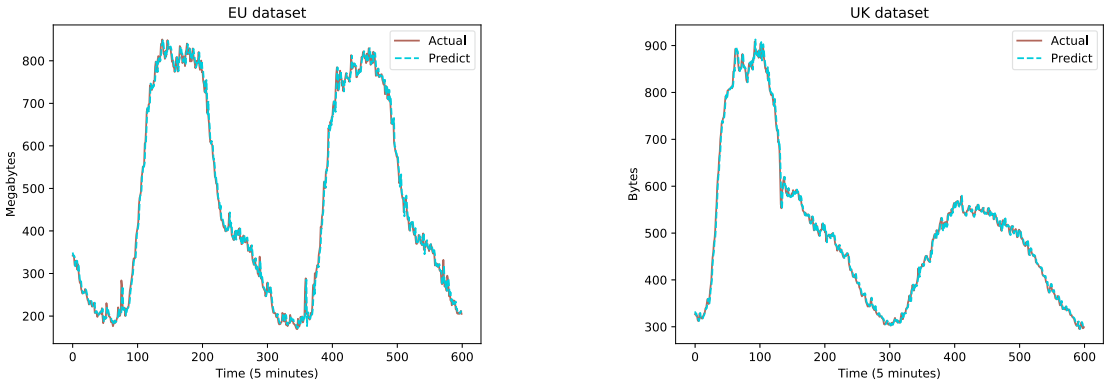

Fig. 2: Visualize the results of OWTO-ELM forecasting for EU (left) and UK (right) dataset

and transformation [22] before being used. In our tested models, we use Root Mean Square Error (RMSE) to evaluate the error between output of traditional network and true value, and the fitness function in meta-heuristic algorithms. For more objective, we also compare the gained results using Mean Absolute Percentage Error (MAPE) and Coefficient of Determination (R^2).

Setting Parameters: The train:test ratio is set to 0.8:0.2 for all datasets

1. Default model settings: The number of layers of MLNN, ELM and versions of ELMs are set to three layers (1 input, 1 hidden and 1 output); The input size $p = km$, where k is the sliding window size; m is dataset metric number (univariate $m = 1$, multivariate $m > 1$); The size of the hidden layer is set to 20 neurons for all models. This setting is chosen small enough to avoid over-fitting and big enough to well-learning the mapping function from inputs to outputs. The output size w is 1. Activation function is set to Exponential Linear Unit (ELU) [4].
2. Meta-heuristics algorithms: For all meta-heuristics algorithms (GA, PSO, TWO, OTWO), we set the population at 20 and the number of generation is 100; GA setting is based on the work [21] with crossover ratio p_c is 0.95 and mutation ratio p_m is 0.025; PSO seting is based on [27]: inertia factor w is set as linearly reducing with iteration from 0.9 to 0.4, cognitive learning rate $c1 = c2 = 1.2$; TWO and OTWO have similar parameters, which are set based on the original paper [**?**] for Eq. 4: $\alpha = 0.99$, scaling factor $\beta = 0.1$, and $\Delta t = 1$.
3. The sliding window size is set to 2 and 5. The size is 2 that means two data points at time step $(t-2)$ and $(t-1)$, i.e., (x_{t-2}, x_{t-1}) are used to predict data point at time step t. The sliding window at 5 is set in a similar way.

4.1. Prediction Accuracy

Table 1 (left side) shows obtained results when running models with EU and UK datasets with different measurement methods. Meantime, Fig. 2 illustrates the prediction outcomes against the real values of testing models.

With EU dataset. When sliding window $k = 2$, our proposed model achieves the two best results as compared with others with RMSE and R^2 measurements. The two best results are 15.363 and 0.9958 for RMSE and R^2, respectively. For MAPE, OTWO-ELM achieves the second-best result that is only after TWO-ELM. When $k = 5$, OTWO-ELM gives the best results in MAPE and R^2 metrics with values of 2.853 and 0.9959, respectively. With RMSE, OTWO-ELM gives the second-best result after PSO-ELM.Hence, it can be made an observation as follows: with all measurement metrics, the obtained results when running MLNN are bad. ELM is slightly better than MLNN. Thus, the values of that metrics are significantly better for combination models among meta-heuristic algorithms and ELM. The outcomes from traditional TWO are quite similar when compared with GA and PSO techniques. However, when combining Opposition-based learning into TWO, the novel OTWO bring the best results as compared with the others in most experiments.

With UK dataset. For OTWO-ELM model, the obtained results show even higher performance in comparison with EU dataset tests. While $k = 2$, OTWO-ELM makes the best results for all metrics including RMSE, MAPE, and

Table 1: Prediction accuracy and the stability between our proposed OTWO-ELM and other models on different datasets

Data	Model	Prediction Accuracy						Stability (based on MAPE)				
		k=2			k=5			Statistics (15 runtimes)				
		RMSE	MAPE	R^2	RMSE	MAPE	R^2	*min*	*max*	*mean*	*std*	*cv*
EU	MLNN	16.716	3.0125	0.9915	15.857	2.9826	0.9927	2.869	3.157	2.965	0.108	0.036
	ELM	16.292	2.9834	0.9948	15.744	2.9255	0.9951	2.882	3.189	2.919	0.077	0.027
	GA-ELM	15.378	2.9203	0.9956	14.914	2.8591	0.9958	2.836	3.054	2.882	0.064	0.022
	PSO-ELM	15.394	2.9255	0.9955	**14.834**	2.8856	0.9959	2.837	2.935	2.863	0.029	0.010
	TWO-ELM	15.400	**2.9171**	0.9955	14.885	2.8645	0.9958	2.843	2.883	2.859	**0.011**	**0.004**
	OTWO-ELM	**15.363**	2.9249	**0.9958**	14.874	**2.853**	**0.9959**	**2.834**	**2.873**	**2.854**	**0.011**	**0.004**
UK	MLNN	11.183	1.5687	0.9958	11.925	1.5945	0.9966	1.395	1.737	1.487	0.108	0.073
	ELM	10.511	1.449	10.190	0.9974	1.4119	0.9975	1.395	1.652	1.460	0.069	0.047
	GA-ELM	10.716	1.4157	0.9975	10.364	1.3399	0.9977	1.330	1.398	1.353	0.025	0.01
	PSO-ELM	10.696	1.4126	0.9975	**10.112**	1.3701	0.9976	1.327	1.402	1.347	0.026	0.019
	TWO-ELM	10.714	1.4170	0.9975	10.410	1.3781	0.9976	1.329	1.385	1.340	0.017	0.013
	OTWO-ELM	**10.349**	**1.3677**	**0.9976**	10.115	**1.3383**	**0.9977**	**1.324**	**1.361**	**1.337**	**0.009**	**0.007**

R^2. Otherwise, when k = 5, OTWO-ELM brings the best results with MAPE, R^2 measurements, and gives the second-best result just after PSO-ELM with RMSE. Note that the difference between RMSE's best result and our proposed OTWO-ELM is very small but there is a huge difference as compared with others. There is also an observation here: the experimental results got with this data is like to EU data. Hybrid models of TWO-ELM, GA-ELM, PSO-ELM produce quite similar results. However, the proposed model OTWO-ELM has better outcomes as compared with other models.

4.2. Stability

Table 1 shows Min, Max, Mean, Standard deviation (Stdev) and Coefficient of Variation (CV) of MAPE values after 15 runtimes with both test data. There are some remarks from the experiments. With UK dataset, the values of Min and Max of OTWO-ELM are the smallest. It means that in the best of worst cases, the OTWO-ELM offers the best results as compared to the others. The Mean values of OTWO-ELM are also the smallest value. The minimum Stdev and CV values also indicate that OTWO-ELM model is the most stable. With EU dataset, it can be seen that OTWO-ELM gives the best results in all 5 values Min, Max, Mean, Stdev. This again proves that our proposed OTWO-ELM outperform in both performance and stability perspectives.

5. Conclusion and Future Work

In this paper, the novel prediction model (OTWO-ELM) for time series data is proposed and validated with experiments. The main contributions of the work include 1) developing a novel version of tug of war optimization based on opposition-based learning; 2) proposing a new prediction model for time series data using a novel variants of extreme learning machine; 3) carrying out experiments and evaluations for OTWO-ELM with real datasets of Internet traffic. The advantages of the proposed model is not only the simplicity, but also its good accuracy, stability, minimal computation cost and interpretable as compared with others. In the future, the approach will be enhanced with more core methods to train different neural network types.

Acknowledgements

The research is funded by the Hanoi University of Science and Technology under project number T2018-PC-017. It is also supported by the projects VEGA 2/0125/20 and APVV-17-0619.

References

[1] Ahmad, A., Javaid, N., Guizani, M., Alrajeh, N., Khan, Z.A., 2016. An accurate and fast converging short-term load forecasting model for industrial applications in a smart grid. IEEE Transactions on Industrial Informatics 13, 2587–2596.

[2] Aljarah, I., Faris, H., Mirjalili, S., 2018. Optimizing connection weights in neural networks using the whale optimization algorithm. Soft Computing 22, 1–15.

[3] Chung, J., Gulcehre, C., Cho, K., Bengio, Y., 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 .

[4] Clevert, D.A., Unterthiner, T., Hochreiter, S., 2015. Fast and accurate deep network learning by exponential linear units (elus). arXiv preprint arXiv:1511.07289 .

[5] Cortez, P., Rio, M., Rocha, M., Sousa, P., 2012. Multi-scale internet traffic forecasting using neural networks and time series methods. Expert Systems 29, 143–155.

[6] Damaševičius, R., Woźniak, M., 2017. State flipping based hyper-heuristic for hybridization of nature inspired algorithms, in: International conference on artificial intelligence and soft computing, Springer. pp. 337–346.

[7] Gal, Y., Ghahramani, Z., 2016. A theoretically grounded application of dropout in recurrent neural networks, in: Advances in neural information processing systems, pp. 1019–1027.

[8] Huang, G.B., Chen, L., Siew, C.K., et al., 2006a. Universal approximation using incremental constructive feedforward networks with random hidden nodes. IEEE Trans. Neural Networks 17, 879–892.

[9] Huang, G.B., Zhou, H., Ding, X., Zhang, R., 2011. Extreme learning machine for regression and multiclass classification. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 42, 513–529.

[10] Huang, G.B., Zhu, Q.Y., Siew, C.K., 2006b. Extreme learning machine: theory and applications. Neurocomputing 70, 489–501.

[11] Kaveh, A., 2017. Tug of war optimization, in: Advances in Metaheuristic Algorithms for Optimal Design of Structures. Springer, pp. 451–487.

[12] Kaveh, A., Shokohi, F., 2016. Optimum design of laterally-supported castellated beams using tug of war optimization algorithm. Structural Engineering and Mechanics 58, 2016.

[13] Kaveh, A., Zolghadr, A., 2017. Guided modal strain energy-based approach for structural damage identification using tug-of-war optimization algorithm. Journal of Computing in Civil Engineering 31, 04017016.

[14] Ma, X., Tao, Z., Wang, Y., Yu, H., Wang, Y., 2015. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. Transportation Research Part C: Emerging Technologies 54, 187–197.

[15] Mahdavi, S., Rahnamayan, S., Deb, K., 2018. Opposition based learning: A literature review. Swarm and evolutionary computation 39, 1–23.

[16] Mirjalili, S., Lewis, A., 2016. The whale optimization algorithm. Advances in engineering software 95, 51–67.

[17] Nguyen, A., Yosinski, J., Clune, J., 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 427–436.

[18] Nguyen, T., 2020a. Opfunu: Python library for optimization functions in numpy. doi:10.5281/zenodo.3620961.

[19] Nguyen, T., 2020b. Otwo-elm: Opposition-based tug of war optimization - extreme learning machine. doi:10.5281/zenodo.3626115.

[20] Nguyen, T., Nguyen, B.M., Nguyen, G., 2019a. Building resource auto-scaler with functional-link neural network and adaptive bacterial foraging optimization, in: International Conference on Theory and Applications of Models of Computation, Springer. pp. 501–517.

[21] Nguyen, T., Nguyen, T., Nguyen, B.M., Nguyen, G., 2019b. Efficient time-series forecasting using neural network and opposition-based coral reefs optimization. International Journal of Computational Intelligence Systems 12, 1144–1161. doi:10.2991/ijcis.d.190930.003.

[22] Nguyen, T., Tran, N., Nguyen, B.M., Nguyen, G., 2018. A resource usage prediction system using functional-link and genetic algorithm neural network for multivariate cloud metrics, in: 11th Conference on Service-Oriented Computing and Applications (SOCA), IEEE. pp. 49–56.

[23] Oliveira, T.P., Barbar, J.S., Soares, A.S., 2016. Computer network traffic prediction: a comparison between traditional and deep learning neural networks. International Journal of Big Data Intelligence 3, 28–37.

[24] Połap, D., Woźniak, M., et al., 2019. Bio-inspired voice evaluation mechanism. Applied Soft Computing 80, 342–357.

[25] Połap, D., et al., 2017. Polar bear optimization algorithm: Meta-heuristic with fast population movement and dynamic birth and death mechanism. Symmetry 9, 203.

[26] Qin, Y., Song, D., Chen, H., Cheng, W., Jiang, G., Cottrell, G.W., 2017. A dual-stage attention-based recurrent neural network for time series prediction, in: IJCAI-17, pp. 2627–2633. doi:10.24963/ijcai.2017/366.

[27] Salih, S.Q., Alsewari, A.A., 2019. A new algorithm for normal and large-scale optimization problems: Nomadic people optimizer. Neural Computing and Applications , 1–28.

[28] Tizhoosh, H.R., 2005. Opposition-based learning: a new scheme for machine intelligence, IEEE. pp. 695–701.

[29] Walczak, S., 2019. Artificial neural networks, in: Advanced Methodologies and Technologies in Artificial Intelligence, Computer Simulation, and Human-Computer Interaction. IGI Global, pp. 40–53.

[30] Xu, Y., Shu, Y., 2006. Evolutionary extreme learning machine–based on particle swarm optimization, in: International Symposium on Neural Networks, Springer. pp. 644–652.

[31] Yang, H., Yi, J., Zhao, J., Dong, Z., 2013. Extreme learning machine based genetic algorithm and its application in power system economic dispatch. Neurocomputing 102, 154–162.

[32] Zhang, H., Wang, Z., Liu, D., 2014. A comprehensive review of stability analysis of continuous-time recurrent neural networks. IEEE Transactions on Neural Networks and Learning Systems 25, 1229–1262.

[33] Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., Xu, B., 2016. Attention-based bidirectional long short-term memory networks for relation classification, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2), pp. 207–212.

[34] Zhu, Q.Y., Qin, A.K., Suganthan, P.N., Huang, G.B., 2005. Evolutionary extreme learning machine. Pattern recognition 38, 1759–1763.