Cuckoo Optimized Multi Layer Perceptron for Web-Services Classification

Syed Mustafa[1]
[1]Professor & HOD, HKBK College of Engineering, Bengaluru, Karnataka, India.
Email: Syedmustafa_blr@yahoo.com

Abstract: Web services describe a collection of operations that are network-accessible through standardized web protocols and its features are described by using a standard eXtensible Markup Language (XML)-based language. Web Services allows integration of applications with ease and less cost. Web Services faces the problem of classifying the categories from a predefined set. In this paper, the focus is on the classification of Web services for managing using Multi-Layer Perceptron-Neural Network (MLP-NN). The MLPNNs are universal approximators and its performance is dependent on the number of hidden neurons. The training accuracy could also be affected by several other parameters, including the number of layers, the number of training samples, the length of learning period, the choice of neuron activation functions, and the training algorithm. To optimize the number of hidden neurons, numerous techniques have been established in literature, which associate it input and output layer sizes or with the number of training samples. A Cuckoo Search (CS) algorithm is proposed to optimize the structure of the MLPNN for improving web service classification.

Keywords: Cuckoo Search (CS), Multi-Layer Perceptron-Neural Network (MLP-NN), Web services, QWS dataset.

I. Introduction

Service-Oriented Architecture (SOA) is anarchitectural style that advances sharing and reusing software components (i.e., published services). Services are discoverable as service providers publish their services' descriptions in registries. Service purchasers can then discover, select, and invoke or compose these published services to meet their business needs. SOA is not fixed to a particular innovation and does not depend on a specific implementation, in spite of the fact that it is normally actualized utilizing web services, being principally created by the Organization for the Advancement of Structured Information Standards (OASIS) and the World Wide Web Consortium (W3C). As characterized in the OASIS reference model for Service Oriented Architecture, "SOA is a paradigm for arranging and using distributed capacities that may under the control of diverse possession domains. It provides a uniform intends to offer, discover, cooperate with and use abilities to deliver desired impacts predictable with quantifiable preconditions and expectations" [1].

The part stressing "under the control of diverse possession domains" is of a specific hobby. It infers the capacity to utilize services gave by outsiders, which frequently accelerates applications' improvement time in correlation with present since a long time ago established, tightly-coupled, embedded situations. Notwithstanding, when service providers and service purchasers are not inside of the same organization, service descriptions could be the main intends to "communicate". Service shoppers don't need to create or even understand the hidden logic and implementation points of interest of services they utilize. Services abstract frame their basic logic, which implies they don't share anything however a formal get that contains just the information required by service shoppers to figure out if a given service is fitting for their requirements (counting functional and non-functional properties of the service) and the information important to collaborate with the service, for example, service interfaces, behavior, and location.

Web service is a software component that takes the input data and produces the output data. Web services are loosely coupling that permits engineers to make, produce and compose them at runtime, interfaces that describe a gathering of operations that are network-open through standardized web protocols and its features are described by utilizing a standard eXtensible Markup Language (XML) based language. Then again, Web Services are linguistically for the most part and described with standards, for example, Simple Object Access Protocol (SOAP), Web Service Description Language (WSDL) and Universal Description Discovery and Integration (UDDI)). The description of web service comprises of the specialized parameters, requirements and strategies that characterize the terms to conjuring service. A web service is characterized as a four-tuple WS = (name, des, In, Out), name speaks to service name and is utilized as a unique identifier; des speaks to the description of service; In speaks to the input parameters set of service; and Out speaks to the output parameters set of service [2].

The SOAP is a protocol trading structured information in a decentralized environment. Web Service Description Language (WSDL) is a XML format for depicting the web services and just describes the syntactic interface of web services that single-handedly can't be utilized for automatic structure of web services. Along these lines, semantic standard protocols, for example, Web Service Description Language-Semantic (WSDL-S), Web Service Modeling Ontology (WSMO), Ontology Web Language-Service (OWL-S) and Semantic Annotations for Web Service Description Language (SAWSDL) have been produced for the automatic web service arrangement and likewise UDDI is a virtual registry that uncovered information about Web services. As a rule, nuclear web service isn't adequate to accomplishing complex needs of the client. Along these lines, web services structure is proper answer for discovering an optimal creation of web services to fulfill different client demands utilizing their syntactic and / or semantic features.

In the field of web services, registries, for example, the Universal Description, Discovery and Integration (UDDI) have been made to empower interoperability and selection of web services. Shockingly, UDDI registries have some real imperfections. Specifically, UDDI registries either are publicly accessible and contain numerous out of date sections or require registration that cutoff points access. In either case, a registry just stores a constrained description of the accessible services. Ontologies made for classifying and using web services can serve as an option arrangement. Be that as it may, the expanding number of accessible web services makes it hard to classify web services utilizing a solitary domain metaphysics or a set of existing ontologies made for different purposes. Moreover, consistent expansion in the quantity of web services requires ceaseless manual push to develop a cosmology [3].

Web Service metrics are exceptionally helpful for designers, providers, and clients. Designers will utilize metrics to manage and control the most ideal approach to actualize the system. Service providers can utilize metrics to ensure the service is running as indicated by certain set measures, and clients can utilize certain metrics to pick the best service providers. So metrics might contrast in view of who is utilizing them, and they might have diverse necessity and level of significance. Web services metrics can be arranged into two fundamental classes: Structural Metrics and Quality Metrics [4].

Structural metrics is essentially about diverse sorts of coupling metric. Coupling is the degree of collaboration in the middle of services, and the essential thought of coupling metric is totally what number of associations there are in the middle of services. The Web Services Quality Metrics primarily allude to both functional and additionally non-functional quality parts of Web Services. This includes performance, reliability, integrity, accessibility, availability, interoperability, and security.

In NN learning, including more hidden neurons is equal to including more premise capacities in capacity approximation. Notwithstanding the quantity of hidden neurons, the training accuracy could likewise be influenced by a few different parameters, including the quantity of layers, the quantity of training samples, the length of learning period, the decision of neuron activation capacities, and the training algorithm. Past work has demonstrated that NNs can be utilized as general approximators. For general approximators, how to decide the correct parameters to use in the model without a preset focus for training accuracy is one of the significant difficulties, which makes the design and utilization of NNs a greater amount of a workmanship than a science [5].

Keeping in mind the end goal to optimize the quantity of hidden neurons, a few methods have been created in writing, which associate it with the quantity of training samples or the input and output layer sizes. Other work assesses the multifaceted nature of the desired capacity and relates it to the quantity of hidden neurons. In the event that the NN training utilizes Back Propagation (BP) algorithm, it has been demonstrated that expanding the quantity of hidden neurons and the quantity of weights makes it less demanding to locate the global minimum. Be that as it may, without looking at the goodness-of-fit or considering the statistical qualities of the training data, these methodologies are less hypothetically stable. Geometric translation provides some knowledge into the issue of deciding the quantity of neurons. It finds the minimum structure of MLP important for an acceptable approximation of a given issue. On the other hand, such system can be just connected to issues with the input space's dimensionality up to two.

Streamlining is a procedure of searching for the optimal answers for a specific issue of interest, and this search procedure can be completed utilizing different agents which basically shape a system of advancing agents. This system can develop by cycles as per a set of principles or mathematical equations. Hence, such a system will demonstrate some developing attributes, prompting self-arranging states which relate to some optima in the search space. Once the self-sorted out states are come to, we say the system converges. In this way, design of a productive advancement algorithm is equal to copying the development of a self-sorting out system [6].

A swarm is characterized as countless or different organisms. Swarm behavior is an aggregate behavior displayed by the swarm with the capacity to communicate specifically or in a roundabout way with one another. The swarm altogether brings out a distributed critical thinking through the swarm behavior. Swarm intelligence is the order that arrangements with natural and artificial systems composed of numerous individuals that facilitate utilizing decentralization and self-organization. The individual organisms in the swarm take after exceptionally basic tenets with no focal control. Indeed, even random correspondence between the organisms in the swarm in the long run results in a savvy general behavior [7].

Amid most recent couple of years, numerous nature motivated developmental algorithms have been created for optimization. These algorithms chip away at the basis of random search in some suitable search region relying upon the issue. Despite the fact that it is a random search, it is not truly random on

the grounds that there is an instrument in the algorithm which controls the search in such manner, that the solution vector gets enhanced regulated. Two vital qualities of these cutting edge meta-heuristics are intensification (exploitation) and diversification (exploration) [8].

Intensification plans to search around the ebb and flow best solutions, while diversification tries to investigate the search space efficiently so that the algorithm does not get stuck into local optimum. Such algorithms have turned out to be entirely well known and causing because of their effectiveness as far as robustness, accuracy, speed and straightforward execution. In any case, in the meantime, they have a few disadvantages like, one specific algorithm may proficient for a particular class of optimization issues however may not be so effective for some different class of optimization issues or sometimes they get stuck into local optimum. One of such nature propelled algorithms is CS algorithm.

In this paper, the structure of the MLPNN is optimized using CS to improve the classification of the web services. The rest of the paper is organized as follows: section 2 presents some relevant works available, section 3 details the methodology, section 4 presents the results and section 5 concludes the paper.

II. Literature Review

Nawi *et al.*, [9] proposed another meta-heuristic search algorithm, called CS, in light of cuckoo bird's conduct to be prepared BP in accomplishing quick convergence rate and to be maintained a strategic distance from local minima issue. The performance of the proposed CS Back-Propagation (CSBP) was contrasted and artificial bee colony utilizing BP algorithm, and other hybrid variants. Particularly OR and XOR datasets are utilized. The reproduction results demonstrate that the computational proficiency of BP training procedure is exceedingly upgraded when coupled with the proposed hybrid technique.

Leung *et al.*, [10] presented the tuning of the structure and parameters of a MLPNN using an improved Genetic Algorithm (GA). It will also be shown that the improved GA performs better than the standard GA based on some benchmark test functions. A MLPNN with switches introduced to its links is proposed.

Xiao *et al.*, [11] applied a Good Points Set-Evolutionary Strategy (GPSES) to solve tuning of both network structure and constraints of a feedforward NN. Good Point Set (GPS) is a concept in number theory.

Subudhi and Jena [12] exhibited two transformative figuring approaches to be specific Differential Evolution (DE) and Opposition based Differential Evolution (ODE) consolidated with Levenberg Marquardt algorithm have been considered for training the Feed-forward NN (FNN) connected for nonlinear framework identification. Results acquired conceive that the proposed consolidated resistance based ODE-NN way to deal with identification of nonlinear framework shows better model identification accuracy contrasted with DE-NN approach. The above strategy was at long last tried on an one degree of freedom (1DOF) exceedingly nonlinear twin rotor multi-input–multi-yield framework (TRMS) to be confirmed the identification performance.

Yeh [13] proposed another delicate registering system called the without parameter Simplified Swarm Optimization (SSO)-based Artificial Neural Network (ANN), or enhanced SSO for short, likewise to be balanced the weights in ANNs. The technique was an adjustment of the SSO, and tries to be conquer a portion of the disadvantages of SSO. In the analyses, the iSSO was contrasted and five different renowned delicate registering techniques, including the BP algorithm, the hereditary algorithm, the Particle Swarm Optimization (PSO) algorithm, agreeable random learning PSO, and the SSO, and its performance was tried on five acclaimed time-series benchmark data to modify the weights of two ANN models (MLP and single multiplicative neuron model). The exploratory results show that iSSO is vigorous and more productive than the other five algorithms.

III. Methodology

In this section, the Quality of Web Service (QWS) Dataset, MLP-NN and CS algorithm are described.

A. Quality of Web Service (QWS) Dataset

Web services in QWS dataset were classified into 4 classifications like: 1) Platinum (excellent); 2) gold; 3) silver and 4) bronze (low quality). Order depended on WSRF's general quality rating. It was grouped into a particular, order based, web service. Web services functionality separates between different services [14].

Redesigned QWS Dataset Version 2.0 has a set of 2,507 Web services and QWS measurements led in March 2008 utilizing a Web Service Broker (WSB) framework. Each line in the dataset speaks to a Web service and its' relating nine QWS measurements (separated by commas). The initial nine components were QWS metrics measured with multiple Web service benchmark apparatuses more than six-days. QWS values speak to measurements averages gathered amid this period. The last 2 parameters speak to service name and reference to WSDL record.

B. Multi-Layer Perceptron-Neural Network (MLP-NN)

MLP is a prevalently utilized neural network structure. In the MLP-NN, the neurons are grouped into layers. The principal and the last layers are called input and output layers, separately, and the remaining layers are called hidden layers. Commonly, a MLP-NN comprises of an input layer, one or more hidden

layers, and an output layer. The inputs are bolstered into the input layer and get multiplied by interconnection weights as they are gone from the input layer to the first hidden layer. Inside of the first hidden layer, they get summed, and then processed by a nonlinear function (as a rule the hyperbolic tangent). At long last the data is multiplied by interconnection weights and then processed one final time inside of the output layer to deliver the neural network output. NNs can without much of a stretch speak to non-linear connections between input data and output data. Regardless of the possibility that the data is fragmented, NNs can effectively arrange the distinctive data classes caught from the network or other sources [15].

Initialization of weight parameters: Preparing the NN for training process. The NN weight parameters are introduced in order to give a decent beginning stage to training process. The generally utilized strategy for MLP weight initialization is to introduce the weights with little scope of random values. Another system propose that the scope of random weights be in aversely relative to the square root of number of jolts a neuron gets on average rate. To enhance the convergence of training system, one can utilize an assortment of circulations and / or diverse reaches and distinctive difference for the random number generators utilized as a part of initialization process of the ANN weights [16].

Making of training process: The most imperative stride in neural model advancement is the NN training process as it gives the information to future execution of the NN model for particular applications. The training data comprises of test sets and a few vectors speaking to the inputs and expected outputs of the NN model. The NN training process can be classified into batch mode training and test by test training. In test by test training, likewise called online training. In batch mode training, otherwise called offline training, weight and epoch is characterize as a phase of the training process that includes presentation of all the training data to the NN once.

MLP-NNs are by and large prepared utilizing the "Back Propagation (BP) algorithm" which is a gradient based supervised learning system. As per the BP algorithm, "a mean squared error between the anticipated and target values for the given input parameters is proliferated in reverse to alter the interconnection between neurons with a specific end goal to minimize the error". An activation function (sigmoid, the tangent hyperbolic, and the linear activation function) in every neuron is mapped as "the total of-weighted input into the activation level" [17].

Owing to its particular structure, a neural network is very good in learning using some learning algorithms such as backpropagation. However, a fixed structure of overall connectivity between neurons may not provide the optimal performance within a given training period. A small network may not provide a good performance due to its limited information processing power. On the other hand, a large network may have some connections redundant. The network structure is considered to be a fully connected network as the number of the hidden nodes is fixed. The number of hidden nodes is physically chosen by increasing it from a trivial number until the learning performance is good enough in terms of fitness value. As a result, a fully connected three-layer network could be reduced to a partially connected network after tuning, as the number of the hidden nodes is fixed. This implies that the cost, in terms of hardware and processing time, of implementing the neural network can be reduced [18].

The real coding technique is applied to solve the problem of tuning the network structure. In the real coding representation, each chromosome is encoded as a vector of floating numbers, with the same length as the vector of decision variables and are optimized using CS.

C. Cuckoo Search (CS) Algorithm

CS is one of the advanced nature enlivened meta-heuristic algorithms. The Greek expressions "meta" and "heuristic" allude to "change" and "discovery arranged by experimentation" separately. Different procedures are utilized to minimize the constraints connected with the issue keeping in mind the end goal to get a global optimum solution.

Cuckoos are alluring birds: The attractiveness is attributable to the delightful sounds created by them furthermore because of their reproduction approach which turns out to be combative in nature. These birds are alluded to as brood parasites as they lay their eggs in communal nests. They evacuate the eggs in the host bird nest with a specific end goal to expand the hatching probability of their own eggs [19].

There are three sorts of brood parasites: the intraspecific brood parasite, cooperation breed and nest assume control sort. The host bird includes in direct combat with the encroaching cuckoo bird. On the off chance that the host bird finds the vicinity of an alien egg, it either discards the egg or forsakes the nest. A few birds are specialized to the point that they have the normal for imitating the shading and the pattern of the egg which diminishes the odds of the egg being forgotten in this manner expanding their productivity.

The timely feeling of egg laying of cuckoo is entirely intriguing. Parasitic cuckoo birds are looking for host bird nests which have recently laid their own eggs. When all is said in done the cuckoo birds lay their eggs sooner than the host bird's eggs with a specific end goal to make space for their own eggs furthermore to guarantee that a large piece of the host bird food is gotten by their chicks.

Levy Flights: In nature, animals hunt down food in a random or quasi-random way. For the most part, the foraging path of an animal is viably a random walk in light of the fact that the following move depends on both the present location / state and the transition probability to the following location. The picked direction verifiably relies on upon a probability, which can be demonstrated mathematically. Different studies have demonstrated that the flight conduct of numerous animals

and creepy crawlies demonstrates the run of the mill qualities of Lévy flights. A Lévy flight is a random walk in which the stride lengths are dispersed by substantial tailed probability distribution. After countless, the separation from the inception of the random walk tends to a stable distribution [20].

CS algorithm depends on the commit brood parasitic conduct of some cuckoo species in combination with the Levy flight conduct of a few birds and fruit flies. A few types of Cuckoo birds lay their eggs in communal nests. On the off chance that a host bird finds the eggs are not their own, they will either discard these alien eggs or just abandon its nest and manufacture another nest somewhere else. CS, can be portrayed utilizing taking after three romanticized rules [25]:

- Each cuckoo lays one egg at a time, and dump its egg in randomly picked nest;
- The best nests with high caliber of eggs will persist to the following generations;
- The number of accessible host nests is altered, and the egg laid by a cuckoo is found by the host conception a probability dad Є [0, 1].

Pseudo code of the CS:

begin
Objective function $f(X)$, $X = (x_1, ..., x_d)^T$
Generate initial population of
n host nests X_i $(i = 1, 2, ..., n)$
while $(t < MaxGeneration)$ *or* (*stop criterion*)
Get a cuckoo randomly by Levy flights
evaluate its quality / fitness F_i
Choose a nest among n (*say*, *j*) *randomly*
if $(F_i > F_j)$,
replace j by the new solution;
end
A fraction (p_a) *of worse nests*
are abandoned and new ones are built;
Keep the best solutions
(*or nests with quality solutions*);
Rank the solutions and find the current best
end while
Postprocess results and visualization
end

While producing new solutions, x (t+1), for the ith cuckoo at iteration (t + 1), the accompanying Lévy flight is performed [22]:

$$x_i^{(t+1)} = x_i^{(t)} + \alpha S(x_i^{(t)} - xbest^{(t)})r$$

Where xbest $^{(t)}$ is the present best solution, α is the stride size parameter, r is a random number from a standard normal distribution, and S is a random walk in view of the Lévy flight. The Lévy flight fundamentally gives a random walk with a stage length drawn from a Lévy distribution. There are a few approaches to apply Lévy flights; in any case, the Mantegna algorithm is the most effective. Consequently in this research, the Mantegna algorithm will be used. In the Mantegna algorithm, the stride length S is figured by $S = \mu / |\nu|^{1/\beta}$

Where β is a parameter between [1, 2], and μ and ν are from normal distribution as:

$$M \sim N(0, \sigma\mu^2) \text{ and } \nu \sim N(0, \sigma\nu^2)$$

$$\text{with } \sigma_\mu = \left(\frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma[(1+\beta)/2]\beta 2^{(\beta-1)/2}} \right)^{1/\beta}, \sigma_\nu = 1$$

IV. Results

The experiments are conducted using QWS data set. The proposed structure optimization with CS was compared with MLP-BPP and MLP-BPP Parameter optimization using CS.

Table I: Summary of Results

	MLP-BPP	MLP-Back Propagation Parameter Optimization Using CS	MLP-Structure Optimization Using CS
Classification accuracy	96.99	96.99	96.99
Avg Precision	0.9701	0.9811	0.9891
Avg Recall	0.97	0.9808	0.989
RMSE	0.1328	0.1015	0.0972

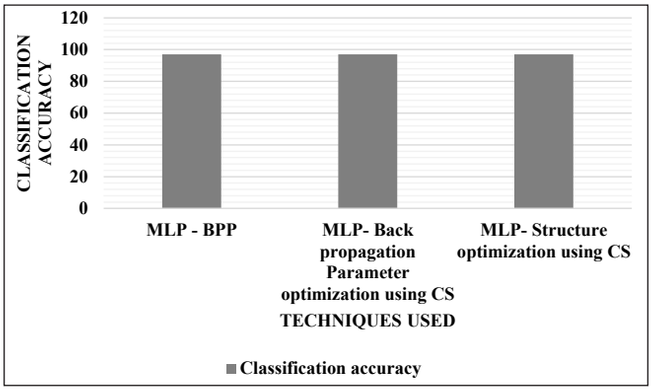

Fig. 1: Classification Accuracy

From the Table I and Fig. 1, it can be observed that the classification accuracy values performs better by MLP-PP, MLP-Back propagation Parameter optimization using CS and MLP-Structure optimization using CS methods.

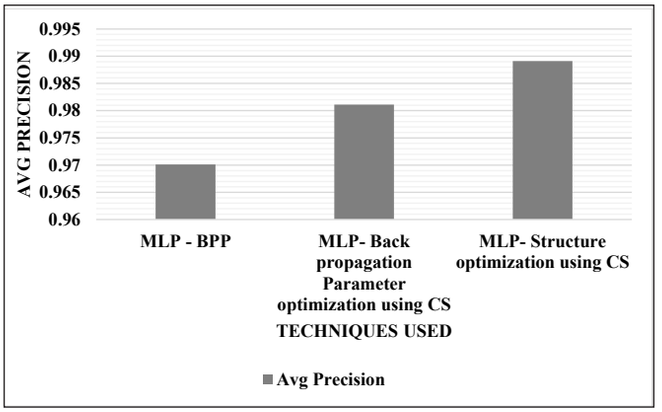

Fig. 2: Avg Precision

From the Table I and Fig. 2, it can be observed that the MLP-Structure optimization using CS method increased avg precision by 1.93% & 0.81% when compared with MLP-BPP and MLP-Back propagation Parameter optimization using CS methods.

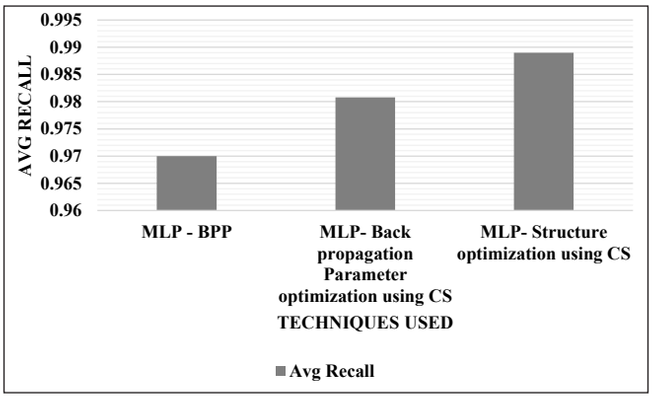

Fig. 3: Avg Recall

From the Table I and Fig. 3, it can be observed that the MLP-Structure optimization using CS method increased avg precision by 1.93% & 0.83% when compared with MLP-BPP and MLP-Back propagation Parameter optimization using CS methods.

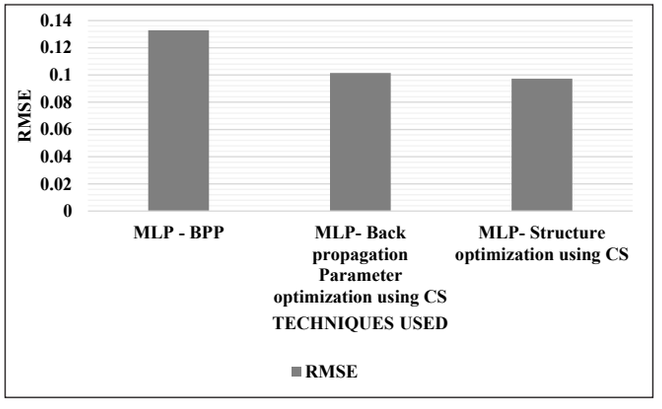

Fig. 4: RMSE

From the Table I and Fig. 4, it can be observed that the MLP-Structure optimization using CS method decreased RMSE by 30.95% & 0.09% when compared with MLP-BPP and MLP-Back propagation Parameter optimization using CS methods.

V. Conclusion

A Web administration is an interface characterizing a collection of operations which are network available through standardized XML messaging. This work concentrates on the classification of medical Web services for overseeing medical services. The proposed CS optimization system concentrated on feature selection and weight optimization for the MLPNN and BBNN classifier. Results demonstrated the proposed strategy's efficiency. The MLP-Structure optimization utilizing CS system diminished RMSE by 30.95% and 0.09% when contrasted and MLP-BPP and MLP-Back propagation Parameter optimization utilizing CS techniques. The MLP-Structure optimization utilizing CS strategy expanded avg precision by 1.93% and 0.81% when contrasted and MLP-BPP and MLP-Back propagation Parameter optimization using CS methods.

References

[1] H. Becha, and D. Amyot, "Non-functional properties in service oriented architecture - A consumer's perspective," *Journal of Software*, vol. 7, no. 3, pp. 575-587, March 2012.

[2] N. H. Rostami, E. Kheirkhah, and M. Jalali, "Web services composition methods and techniques: A review," *International Journal of Computer Science, Engineering and Information Technology*, vol.3, no. 6, pp. 15-29, December 2013.

[3] A. Segev, and Q. Z. Sheng, "Bootstrapping ontologies for web services," *IEEE Transactions on Services Computing*, vol. 5, no. 1, pp. 33-44, 2012.

[4] M. I. Ladan, "Web services metrics: A survey and a classification," *Journal of Communication and Computer*, vol. 9, no. 7, pp. 824-829, 2012.

[5] Y. Liu, J. A. Starzyk, and Z. Zhu, "Optimized approximation algorithm in neural networks without overfitting," *IEEE Transactions on Neural Networks*, vol. 19, no. 6, pp. 983-995, 2008.

[6] X. S. Yang, and S. Deb, "Cuckoo search: Recent advances and applications," *Neural Computing and Applications*, vol. 24, no. 1, pp. 169-174, 2014.

[7] N. Kayarvizhy, S. Kanmani, and R. Uthariaraj, "Ann models optimized using swarm intelligence algorithms," *WSEAS Transactions on Computers*, vol. 13, no. 45, pp. 501-519, 2014.

[8] H. Soneji, and R. C. Sanghvi, "Towards the improvement of cuckoo search algorithm," In *2012 World Congress on Information and Communication Technologies (WICT),* IEEE, pp. 878-883, October 2012.

[9] N. M. Nawi, A. Khan, and M. Z. Rehman, "A new back-propagation neural network optimized with cuckoo search algorithm," In *Computational Science and its Applications-ICCSA 2013,* Springer Berlin Heidelberg, pp. 413-426, 2013.

[10] F. H. Leung, H. K. Lam, S. H. Ling, and P. K. Tam, "Tuning of the structure and parameters of a neural network using an improved genetic algorithm," *IEEE Transactions on Neural Networks,* vol. 14, no. 1, pp. 79-88, 2003.

[11] C. Xiao, Z. Cai, Y. Wang, and X. Liu, "Tuning of the structure and parameters of a neural network using a good points set evolutionary strategy," In *The 9th International Conference for Young Computer Scientists, 2008, ICYCS 2008*, IEEE, pp. 1749-1754, November 2008.

[12] B. Subudhi, and D. Jena, "A differential evolution based neural network approach to nonlinear system identification," *Applied Soft Computing*, vol. 11, no. 1, pp. 861-871, 2011.

[13] W. C. Yeh, "New parameter-free simplified swarm optimization for artificial neural network training and its application in the prediction of time series," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 4, pp. 661-665, 2013.

[14] A. S. Mustafa, and Y. S. Kumaraswamy, "Performance evaluation of web-services classification," *Indian Journal of Science and Technology*, vol. 7, no. 10, pp. 1674-1681, 2014.

[15] L. M. Ibrahim, "Anomaly network intrusion detection system based on Distributed Time-Delay Neural Network (DTDNN)," *Journal of Engineering Science and Technology*, vol. 5, no. 4, pp. 457-471, 2010.

[16] M. Chakraborty, "Artificial neural network for performance modeling and optimization of CMOS analog circuits," arXiv:1212.0215, 2012.

[17] W. Duch, and J. Korczak, "Optimization and global minimization methods suitable for neural networks," *Neural Computing Surveys 2*, pp. 163-212, 1998.

[18] J. T. Tsai, J. H. Chou, and T. K. Liu, "Tuning the structure and parameters of a neural network by using hybrid Taguchi-genetic algorithm," *IEEE Transactions on Neural Networks*, vol. 17, no. 1, pp. 69-80, 2006.

[19] A. S. Charan, N. K. Manasa, and N. V. S. N. Sarma, "Uniformly spaced planar antenna array optimization using cuckoo search algorithm," *Computer Science & Information Technology (CS & IT)*, pp. 157-167, 2014.

[20] E. Valian, S. Mohanna, and S. Tavakoli, "Improved cuckoo search algorithm for feedforward neural network training," *International Journal of Artificial Intelligence & Applications*, vol. 2, no. 3, pp. 36-43, 2011.

[21] S. Kamat, and A. G. Karegowda, "A brief survey on cuckoo search applications," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 2, no. 2 special, 2014.

[22] J. F. Chen, Q. H. Do, and H. N. Hsieh, "Training artificial neural networks by a hybrid PSO-CS algorithm," *Algorithms*, vol. 8, no. 2, pp. 292-308, 2015.