JOURNAL OF PURE AND APPLIED MICROBIOLOGY, Aug. 2015. Vol. **9**, p. 169-181
SPECIAL ISSUE ON RECENT RESEARCH CHALLENGES IN BIO-MEDICAL APPLICATIONS

Mobility Prediction Using Modified Neural Network For Integrated Disaster Management

J. Ananthi[1] and V. Ranganathan[2]

[1]ECE Department, Dhanalakshmi College of Engineering, Chennai, India.
[2]Department of ECE, SreeSastha Institute of Engineering and Technology, India.

(Received: 20 February 2015; accepted: 25 April 2015)

During natural disasters like earth quake and floods, various public emergency agencies like fire services, civil defenses have to coordinate at the disaster site without proper infrastructure. To achieve communication at ground zero wireless networks play a very important role. The wireless network provides global services through integrated networks and using multihop communication without infrastructure. To achieve Quality of Service (QoS) Mobility prediction, precise and competent forecast of mobile users trail is of significance for improved network performance. Mobility prediction along with wireless communication protocols helps in resource management in a disaster management scenario. In this paper a mobility prediction mechanism using Neural Network is proposed. A novel training algorithm based on parallel Swarm Intelligence algorithm is proposed. The proposed technique is evaluated on Multi-Layer Perceptron Neural Network (MLPNN) and Jordan Network. Simulation results on large mobility traces show high degree of classification accuracy.

Key words: Mobility Prediction, Bee Algorithm (BA), Fish School Search (FSS), Multi-Layer Perceptron Neural Network (MLPNN) and Jordan Network.

Wireless network enables communication without the need of wire and hence provides mobility to the user. Wireless networks can be broadly classified into fixed infrastructure based system which have Access Point (AP) to connect to the external world. The other type of wireless network is the infrastructure less network or Ad hoc network which do not have any fixed infrastructure. A third type of network called hybrid network is emerging using the strengths of both types of networks. Wireless networks play a crucial role for communication when disaster strikes unexpectedly. With a well-designed hybrid wireless network, monitoring becomes easy with minimal infrastructure. Figure 1 shows a typical wireless network with temporary AP during disaster. Understanding network traffic behaviour is paramount in evolution of wireless networks. This results in efficient network bandwidth resources planning and management. Early radio resource reservations in future user mobile trajectory locations ensure optimized network resource allocation due to it being limited and to ensure sustained Quality-Of-Service (QoS) levels. This makes sure that network services are available anywhere/anytime, which becomes possible only when users demands are at a given time can be predicted. Wireless network activity, and users will increase with slow but sure improvement of wireless applications requiring higher bandwidth[1]. The trend of anywhere/anytime wireless access lead to users anticipating improved network supported QoS levels. Hence, accurate and efficient users travel path or mobility prediction is greatly important for overall network performance.

Accurate mobility prediction provides smaller data-dropping probability and reduced handover latency in cellular network and infrastructure based network. For easy mobility

* To whom all correspondence should be addressed.
E-mail: ananthi_research@rediffmail.com

prediction algorithm used in wireless environments must control overhead, be knowledgeable about geographical areas, be user intensive and be adaptable to regular/random user moves. Velocity and positional co-ordinates ensure user mobility predictions with regard to cell structure, through location techniques like GPS. But user's earlier movement patterns in Mobility History Bases (MHB) can be exploited or stochastic models can be taken recourse to.

Wireless networks are divided into geographical units called cells, each cell providing wireless coverage and being administered through a single access point/base station. Sometimes, mobile users could need on-going connections transferred between base stations to ensure active sessions, a process called handoff. Successful handoffs become possible when sufficient resources are given to an on-going session by a new network access point. Else, the session is prematurely terminated/dropped due to a new cell lacking resources.

It is known that node behavior reveals patterns in mobility prediction, as it is impossible to predict future network state through random node behavior. A wide pattern range is observed ranging from a possibility of network location in an office building with nodes moving along corridors and stopping in offices, or a network location on an expressway where nodes only move along the street. Such patterns ensure current node behavior mapping as also its future state. User movement trajectories are generally logged in at the time when a mobile device is connected to an Access Point (AP), which represents a specific AP of the nth user location has moved from defined time. User mobility pattern (UMP) is the frequently used user path and is got through using logs from all APs. User mobility patterns generate mobility rules.

Recent years witnessed extensive work on mobility prediction schemes development. Most models rely on historical data including information on aggregate mobility and every location's handoff history[2,4]. But some models use both mobility historical data and current network conditions. An example is the model featured in Akyildiz and Wang[5] that considers both velocity and mobile user's direction as also historical data to predict future locations. In [6], a hierarchical location prediction model employing mobility history to predict inter-cell movements while simultaneously considering mobile users speed and movement direction within a cell was proposed. This was limited to location prediction alone.

Many studies about the mobility of users in wireless environments give the information that there is some regularity in the movement of mobile users. Such regularity in the mobility patterns can be recorded as a profile of an individual user and can be used for designing a prediction algorithm[7]. Profile based prediction algorithm is useful for next cell prediction and signal strength prediction based on the user profile and the location classifications. Designing a proper prediction scheme avoids incorrect handoff requests and makes destination driven handoff patterns for mobile stations. A neural network based prediction scheme can be designed by using user information and the path travelled by the user in a cell. Poon and Chan[8] proposed a neural network based predictor with two hidden layers by using the distance from the base station and the user, and moving direction of the user as inputs. This predictor predicts the next cell and reserves the needed resources in a predicted cell. This neural network also used the prior knowledge about the geographical structure of cellular networks and the prediction is independent of the user. When there is a disturbance in the movement of the user like sudden stop, moving back and then forwards, then filtering and smoothing is done on the input data. Neural networks are capable to learn the relationship between complex inputs and outputs. During the learning process, the knowledge is acquired and stored as synaptic weights of inter nodal links. The fitness of a NN is measured by the training time and the error (i.e. the difference between actual and expected outcome). To reduce the training time and error rate best set of inter nodal weighs are learned quickly by applying some optimizations.

This paper investigated Artificial Neural Network for mobility prediction. For neural networks to predict accurately, training algorithms play a crucial role. Standard Back Propagation algorithm and Back Propagation Through Time (for recurrent neural networks) do not perform well for large networks with high computational effort and is NP- Complete. To overcome the NP-Complete problem metaheuristic algorithms including Genetic Algorithm, Particle Swarm Optimization have been used as a learning algorithm. In this work a new training algorithm based on hybrid metaheuristic algorithm is proposed. The proposed algorithm is based on using the popular Bee Swarm and Fish Swarm Algorithm in parallel. The algorithm was adapted for MLP Neural Network and Jordan Neural Network. Dartmouth College's month long trace data in the public domain was used to evaluate the proposed method. The rest of the paper is organized in sections dealing with related works, proposed methodology, experimental setup and results.

Related Works

Akoush andSameh[9] proposed a mobility path prediction model and hybrid Bayesian neural network model to predict cellular networks locations, based on probability model to represent relationships uncertainty. Simulation results with realistic mobility patternsshowed that the proposed algorithm achieved higher prediction accuracy.

Kaanicheand Kamoun[10] proposed a recurrent neural network for long-term time series prediction, as mobility prediction is problematic in time series prediction. This neural predictor architecture is a three-layer network with feedback connections and trained by back propagation through Time algorithm. To evaluate this predictor's efficiency in mobility prediction, it was tested on time series describing Ad hoc mobile node moving locations, according to RWM model.

Soriano and Urano[11] proposed a modified random data replication model within a mobile peer-to-peer network based on the mobile node predicting condition to replicate its object. It uses unsupervised learning neural networks called Self Organizing Map to classify each node's input attributes and provides a training set - a basis to identify a nodes' current state.

Abu-Ghazaleh andAlfa[12] applied Markov Renewal Processes to mobility modeling and likelihood predicting of next-cell transition, along with anticipating intra transition duration for arbitrary wireless network user. This prediction technique can also be extended to compute likelihoods of users being in a specific condition after N transitions. The proposed technique also estimates expected spatial-temporal traffic load and location activity in a network area.

Fotouhi, *et al.*,[13] presented a reliable and real-time mobility support in WSNs, which need mandatory handoff and rerouting decisions. A mechanism is informed to heuristics to take reliable WSN handoff decisions. A two-phase procedure performs handoff decision based on several important metrics, using fuzzy logic to combine them.Ling andWang[14] proposed Artificial Bee Colony (ABC) algorithm for optimizing the synaptic weights of ANN, because ANN were used for certain classification and regression problems. The experimental results of ANN and ABC-ANN were compared. Results revealed that ABC-ANN produced high accuracy after training.Saadi, *et al.*,[15] proposed ABC optimization for Neural Network Model to classify ECG Signals. Experiments proved that using ABC with NN reduced the number of control parameters for classification.

Yu, *et al.*,[16] proposed Fish Swarm Optimization for optimizing the ANN to predict/ diagnose the faults in transformers. After conducting experiments, authors concluded that the convergence rate is very fast for NN when FSO was used. Shiand Li[17] proposed Ant Colony Optimization (ACO) for optimizing back propagation neural networks for assessing the performance of residential buildings.Most of the back propagation algorithm gave local optimum weights. When training by samples, ACO-BNN yielded global solutions, because ACO-BNN used dynamic forecasting of errors.

Mourad, *et al.*,[18] proposed ACO for tracking target nodes of mobile sensor nodes network. Some fixed points were selected in the city as a set of positions and current position of a single node was estimated and assigned within the set. ACO algorithm was used to find the new location optimally based on using interval theory.Parija, *et al.*,[19] proposed location prediction of mobile devices of cellular networks by training a multi-layerNN. Past history about the mobility of devices were used as training samples to train the NN. Experiment results showed that performance of trained NN was satisfactory.

METHODOLOGY

In this work we used mobility trace collected over three years by Dartmouth College.

The initial 476 APswere increased to 566 over time when the data was being collected. SSID are the same for all AP's and 115 subnets covered 188 buildings. APsare forced to obtain new IP addresses occasionally. Mobility trace of 5500 students and 1200 faculty were collected over 3 years. A syslog server log included timestamp to each message and has the AP name, card MAC address and message type. Messages are authenticated, associated, reassociated, roamed and disassociated. When a mobile device links a network it is first authenticated, then associates with an AP enabling all device-network traffic. Reassociation is when another AP with better signal strength becomes available. Roaming is when a device re-associates with a new access point. Disassociated message is sent when the device needs the network no more. Sample syslog data is given in Table 1.

The mobility pattern can be used as the input to neural network. A classic Multi-Layer Perceptron (MLP) arrangement consists of source nodes in the input layer, one or more hidden layers which compute the inputs by applying activation function, and an output layer of nodes illustrated in Figure 3. The network has an input layer, single hidden layer with non-linear activation and an output layer with a linear function. The input signal flows through the hidden layer from the input layer to the output layer. The computations performed by the feedforward network can be written mathematically as

t = f (s) =B ö (As +a) +b

s = inputs

t = outputs

A = first layer weight matrix

a = first layer bias vector

B = second layer weight matrix

b = second layer bias vector

ö = non-linearity function.

MLP can estimate any continuous function to any level of accuracy of a compact set. Alternatively, the number of hidden layers and weight matrix that ensure optimum network convergence is never known. The solutions to each neural network for the input and output data applied is unique. An MLP can categorize non-linear problems successfully.The network is governed by equations that steer the network to provide precise result with minimum training error. In General the training algorithms concentrate on synaptic weights, number of hidden layers, activation function etc. for assuring optimal result. The familiar supervised learning technique that trains the neural network is Back Propagation (BP). On the other hand, BP gets locked in local minima and has slow convergence at some time. However to overcome this trapping of local minima some evolutionary algorithm can be used.

Multi-layeredfeed forward neural networks are appropriate for complex pattern classification since it has various characteristics that provide the solution for the same. However, the lack of a suitable training algorithm restricts its application for some of the real world environment. Finding a training algorithm which provides a near global optimal set of parameters for a comparatively short interval of time is a complicated task. Evolutionary algorithms like Particle swarm

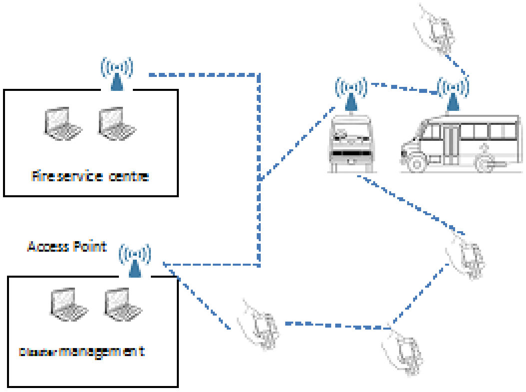

Fig. 1. A disaster management scenario with temporary access point

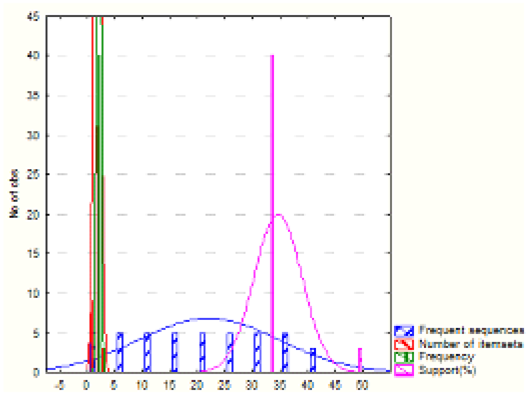

Fig. 2. Frequent Item set for one month data

intelligence and Genetic algorithms explore a large and composite space in an intellectual way to locate parameters nearer to the global optimum. However the training cost of genetic algorithm is quite high.

Another popular Neural Network used for prediction is the Partially recurrent networks. They have a special neurons group in the input layer, called context neurons/neurons of state. Thus,in an input layer of partially recurrent networks two neuron types are seen, those which act like the input, receiving outside signals and context neurons receiving output values of a layer delayed by a step. They are useful for time series prediction problem[21-25]. Jordan in 1986[23, 34] proposed Jordan neural network, characterized as context neurons receive a copy from output neurons and themselves. The Jordan network has as many context neurons as output neurons. Recurrent output layer connections to the context neurons have an associated parameter, m, that, usually take a constant value positive smaller than 1.

For time series prediction, a network will have an output neuron representing predicted time series value at futures instances. The network will thus have only a context neuron and its activations at instant t is given by the following expression:

$$c(t) = mc(t-1) + x(t-1)$$

where x(t-1) is output network at instant t-1.

Remaining network activities arecalculated as in multiplayer perceptron, where it is enough to consider as input vector a concatenation of the external input activations and context neurons activations:

$$u(t) = \left(x(t),\ldots,x(t-d),\ c(t)\right)$$

Taking into account the expression of the context neuron activation, it is possible to write:

$$c(t) = \sum_{j=1}^{t-1} \mu^{j-1} x(t-j)$$

Therefore, parameter m equips Jordan network with certain inertia for this network's context neurons. It was previously seen that context neuron accumulates network output at all previous instants and parameter value m determines context neuron's sensitivity to retain information.

Popular Jordan NN training algorithm include Back Propagation Through Time (BPT). Weight adjustment between back propagation processing elements is carried out based on the difference between neural network's target and output values. Error difference in backpropagation is measured by mean square error, as revealed below:

$$E = \sum_{k=1}^{m} \sum_{j=1}^{q} \left(t_{kj} - z_{kj}\right)^2$$

Wheret_{kj} is the j_{th} target value of the k_{th} compound, and z_{kj} is the output. Weights are adjusted to a gradient direction with better fitness[27] as shown in the equation:

$$w_{ji}^{new} = w_{ji}^{old} + \alpha \sum_{k} \delta_{kj} y_{ki} + \beta \Delta w_{ji}^{old}$$

where j, iare adjacent layer indices, $w_{ji}$is weight from the previous layer i_{th}neuron to the

Table 1. Sample syslog of a user

Unix Time Stamp	Specific Access Point Associated with the User
1035100785	AdmBldg19AP3
1035100842	AdmBldg18AP3
1035100851	AdmBldg35AP1
1035100908	AdmBldg18AP3
1035100963	AdmBldg35AP1
1035101020	AdmBldg18AP3
1035101022	AdmBldg35AP1
1035101080	AdmBldg18AP1
1035101082	AdmBldg35AP1
1035101139	AdmBldg18AP3

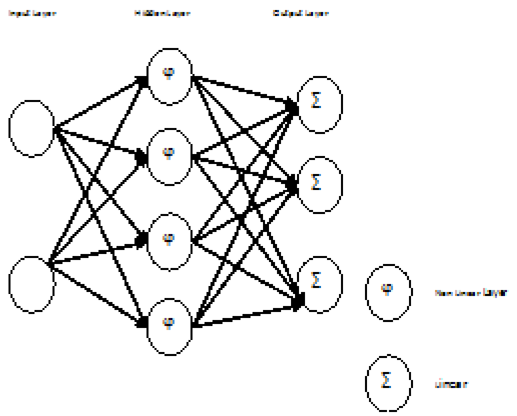

Fig. 3. Multi-layer perceptron architecture

j_{th}neuron in the current layer and Äw_{ji} is the preceding weight change. The variable y_{ki} represents the i^{th} output for the k^{th} pattern. Parameters á and â are positive constants called learning rate and momentum rate which controls weight adjustments amount during weight updation[27].

Back-propagation (BP) algorithm, a gradient based method, is the most commonly used in neural network training. BP algorithm's inherent problems are encountered when this algorithm is used. First, BP algorithm is easily trapped in local minima for non-linearly separable pattern classiûcation problems/complex function approximation problem[28], leading to back-propagation failure to locate a global optimal solution. Second, BPO algorithm's convergent speed is too slow even if learning goal, a given termination error, is achieved. What is to be emphasized is that BP algorithm's convergent behavior depends on initial values choices in network connection weights as also algorithm parameters like learning rate and momentum. To improve original BP algorithm performance, researchers concentrated on two factors:

(1) Better energy function selection[29, 30];

(2) Dynamic learning rate and momentum selection[31, 32].

But these have not removed BP algorithms disadvantages of being trapped in local optima. Specifically, convergent speed will be slower as neural network's structure is more complex.

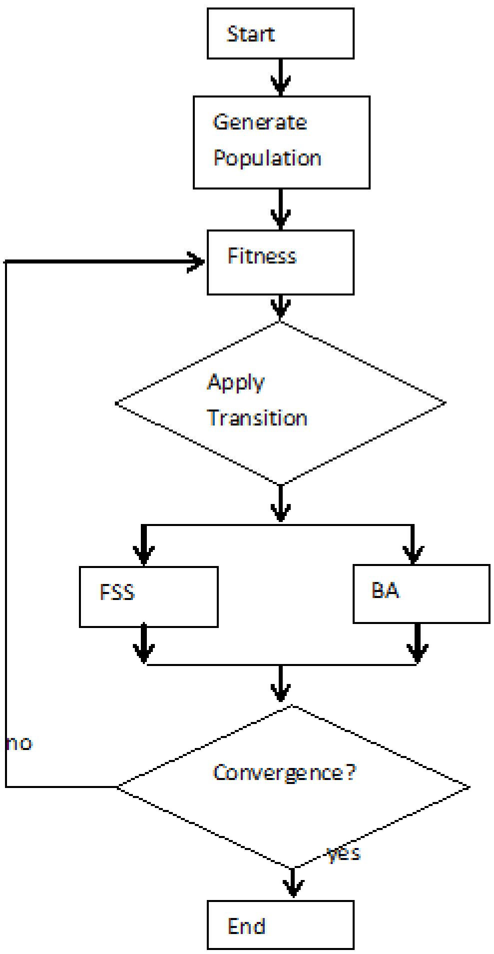

Fig. 4. Flowchart for proposed Hybrid Algorithm

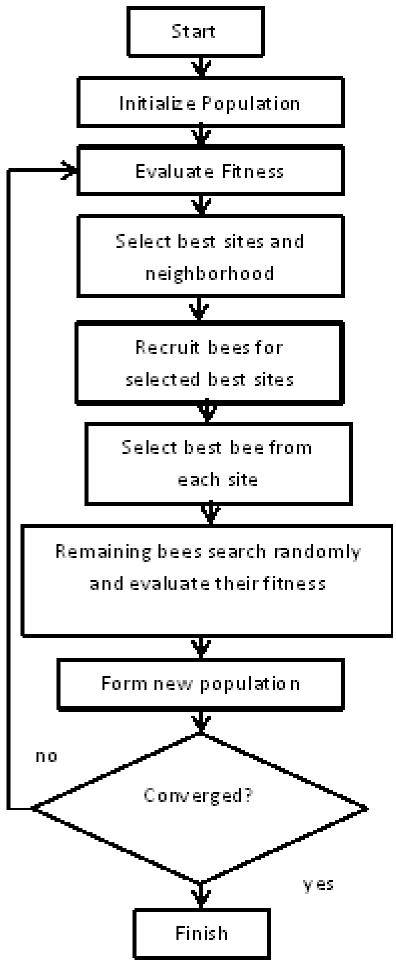

Fig. 5. Flowchart for Bee Algorithm

Proposed Learning Technique –Hybrid meta heuristic algorithm

The basic idea for this hybrid algorithm is searching for an optimum, at accelerated training speed. When ûtness function value is the same for some generations, or value changed is lower than a predeûned number, searching is switched to gradient descending searching based on this heuristic knowledge. From the research in SI, the proposed system hybridizes Fish School Search (FSS) and Bee Algorithm (BA). The two techniques for this include transitional technique and parallel technique. FSS and BA are hybridized using parallel technique in this paper.

1. Select population of size n randomly and Initialize population.
2. Evaluate individual fitness.
3. When a termination criterion is not met, split population to ensure selective reproduction and velocity updating, depending on algorithm used.
4. If algorithm used is BA perform operations. Or else perform personal best calculation.
5. Repeat process till final solution.

In the proposed parallel method, the population is divided into two and evolved with two techniques. The algorithm executes both techniques simultaneously selecting a user specified number of best individuals from each system after exchanging many user defined iterations. Individual with larger fitness value are usually selected. The steps involved in the proposed system is given by:

Table 2. The metaheuristic algorithm parameters

Bee Algorithm	
Initial number of worker bees	10
Initial number of onlooker bees	10
Fish School Search	
Initial number of fishes	10
Visual range	0.5
Step	0.25
Common Criteria	
Total Number of iteration	500
Termination criteria	MSE< 0.1

Table 3. Parameters measured for different experiments.

	True Positive Rate	Positive Predictive Value	F Measure	True Positive Rate	Positive Predictive Value	F Measure
		MLP NN - BP			Jordan NN - BPTT	
Library	0.745	0.8247	0.7828	0.775	0.8304	0.8017
Academic	0.865	0.8251	0.8446	0.8717	0.8546	0.8631
residential	0.81	0.8648	0.8365	0.815	0.8655	0.8395
social	0.8417	0.7842	0.8119	0.8467	0.79	0.8174
Admin	0.8	0.7705	0.785	0.8	0.7742	0.7869
		MLP NN - BA			Jordan NN - BA	
Library	0.8283	0.8599	0.8438	0.8567	0.8939	0.8749
Academic	0.8867	0.8567	0.8714	0.895	0.895	0.895
residential	0.8483	0.8791	0.8634	0.86	0.8821	0.8709
social	0.8467	0.8167	0.8314	0.8833	0.836	0.859
Admin	0.8267	0.8267	0.8267	0.8267	0.8185	0.8226
		MLP NN -FSS			Jordan NN - FSS	
Library	0.835	0.8867	0.8601	0.8567	0.8955	0.8757
Academic	0.895	0.8847	0.8898	0.9083	0.9083	0.9083
residential	0.8483	0.8806	0.8641	0.87	0.8832	0.8766
social	0.8833	0.823	0.8521	0.885	0.8523	0.8683
Admin	0.8267	0.8185	0.8226	0.8367	0.8203	0.8284
		MLP NN - HYBRID			Jordan NN - HYBRID	
Library	0.9017	0.9108	0.9062	0.9017	0.9139	0.9078
Academic	0.915	0.9181	0.9165	0.9167	0.9182	0.9174
residential	0.8817	0.8846	0.8831	0.885	0.8985	0.8917
social	0.8933	0.8673	0.8801	0.8933	0.8673	0.8801
Admin	0.8467	0.8581	0.8524	0.8633	0.8633	0.8633

Karabogaand Basturk[35] originally presented the Bee Colony algorithm inspired by the behavior of bees with better performance in function optimization problems compared to GA, Differential Evolution (DE), and Particle Swarm Optimization (PSO)[26]. It is based on bee swarms food foraging behavior. The algorithm in a basic version performs a neighborhood and random search combined and which is used for both combinatorial optimization and functional optimization.

A bee colony can extend itself over long distances and many directions simultaneously exploiting many food sources[35, 36]. Theoretically, a flower patch with plenty of nectar/pollen that is collected with less effort is visited by more bees, whereas patches with reduced nectar/pollen receive fewer bees [36, 37]. Foraging begins in a colony by scout bees scouting for promising flower patches. They move randomly from one patch to another. During harvesting, a colony continues exploration, using some in the bee population as scout bees[35].

When they return to the hive, scout bees locating a patch are rated higher than a specific quality threshold (measured as a combination of constituents like sugar content) deposit nectar/pollen and proceed to the "dance floor" to perform a "waggle dance". This mysterious dance is essential for colony communication, and includes three information pieces regarding a flower patch: where it is located, distance from the hive and its quality rating (or fitness)[35, 37]. This helps the colony to send bees to flower patches precisely. Each

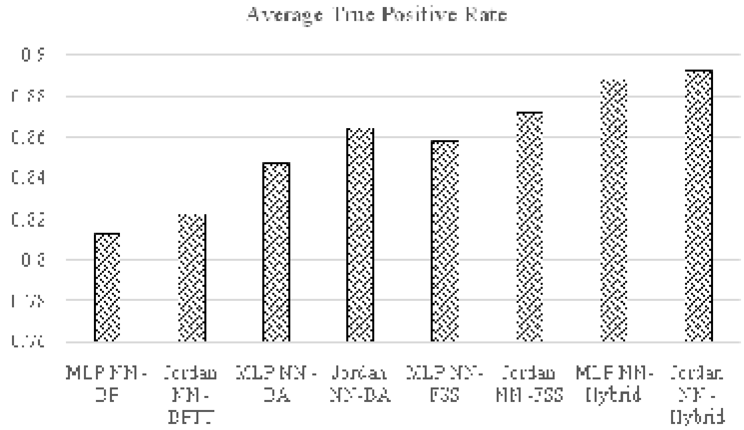

Fig. 6. True Positive Rate

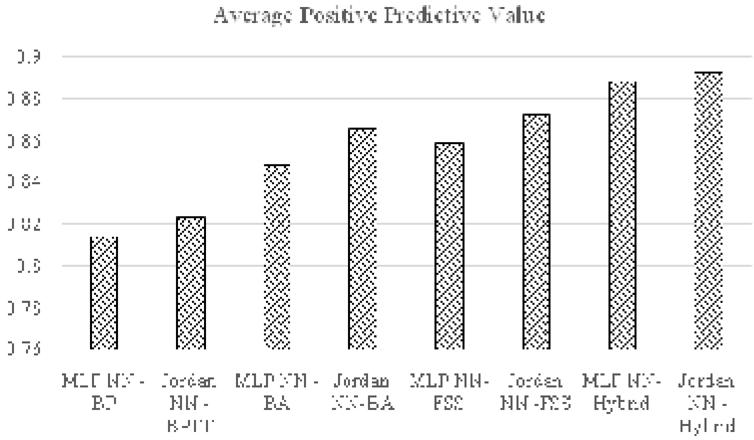

Fig. 7. Positive Predictive Value

individual's outside environment knowledge is gleaned from the waggle dance which enables a colony to evaluate different patches relative merit according to food quality and energy required to harvest it[38]. After waggle dancing, the dancer (i.e. the scout bee) goes to the flower patch with follower bees. Additional follower bees are sent to promising patches allowing a colony to gather food quickly and efficiently.

Bees monitor food level when harvesting from a patch this being necessary to decide the next waggle dance when they return[38]. If the patch is still promising as a food source, it is advertised in the waggle dance and more bees recruited to that source. Figure 5 shows the algorithm's pseudo code in its simplest form. It needs many parameters including scout bees number (n), sites selected from n visited sites (m), best site number out of m selected sites (e), number of bees needed for best e sites (nep), number of bees recruited for other (m-e) selected sites (nsp), initial patch size (ngh) including site, its neighbourhood and stopping criterion. The algorithm starts with n scout bees placed randomly in the search space. Figure 3 shows the flow chart of the Bee Algorithm.

Optimization algorithms solve real-world problems and present ability to deal with dynamic environments, where optima solutions change with time. Another swarm intelligence algorithm based on fish behaviour, the Fish School Search algorithm (FSS)[39, 40], contains an interesting feature useful for dynamic environments.The FSS is an optimization algorithm founded on oceanic fish's gregarious behavior. First proposed by Bastos-Filho, *et al.*,[40], it ensures that each fish represents a solution. Fish's success during search is

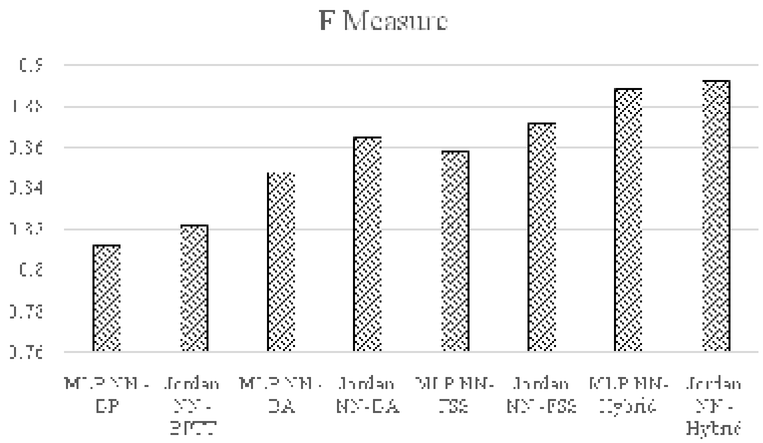

Fig. 8. F Measure

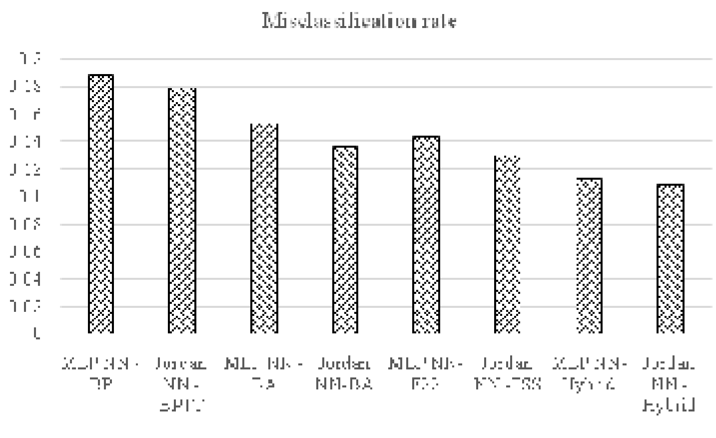

Fig. 9. Misclassification rate

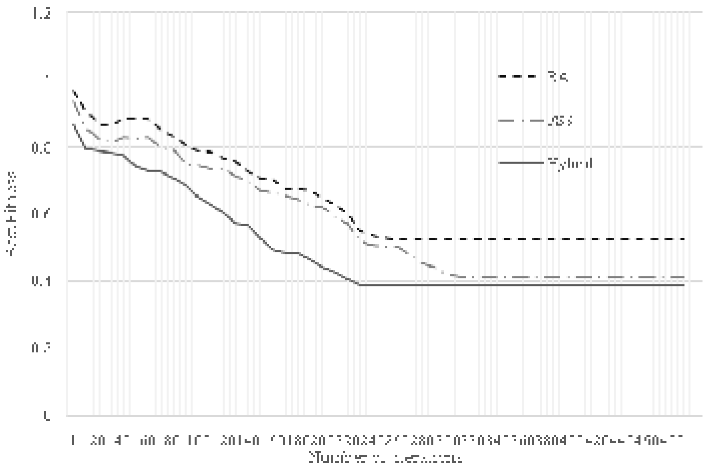

Fig. 10. Best Fitness

indicated by its weight. FSS has 4 operators, executed for each fish at every iteration: (i) individual movement, responsible for local search step; (ii) feeding, which updates fish weights indicating success/failure degree during search; (iii) collective-instinctive movement, ensures that fish move in a specific direction; and (iv) collective-volitive movement, that controls search granularity.

Feeding operator determines fish weight variation at every iteration. A fish can increase or decrease its weight based on the search process success/failure. Fish weight is evaluated according to the following equation:

$$w_i(t+1) = w_i(t) + \frac{\Delta f_i}{\max(|\Delta f|)},$$

where Wi(t) is the weight of the fish i, Δf_i is the fitness function difference between new position and current fish position, max(|Äf|) is the absolute value of the greatest fitness difference among all fish. There is a parameter w_{scale} limiting maximum fish weight. Individual fish weight can vary between 1 and w_{scale}, having an initial value equal to $w_{scale}/2$.

Collective-volitive movement operator controls search granularity of the fish school. When the school achieves better results, operator approximates the fish planning to accelerate convergence toward a good region. On the contrary, the operator disperses the fish away from the school's barycenter with fish now having more chances to escape a local minimum. Fish school expansion/contraction is applied as a small drift to every fish position regarding school barycenter, which is evaluated as shown below:

$$\vec{B}(t) = \frac{\sum_{i=1}^{N} \vec{x}_i(t) W_i(t)}{\sum_{i=1}^{N} \vec{x}_i(t)}$$

Experimental Setup

The mobility prediction accuracy is evaluated usingMulti-Layer Perceptron Neural Network and Jordan neural network. The parameters used in the proposed hybrid algorithm is shown in Table 2.

RESULT AND DISCUSSION

Experiments were conducted using 3000 associations obtained from the syslog dataset. The class label is the head of the associated outcome. Five classes namely Academic area, residential area, library, social and administrative buildings were considered as the location of the mobile device in the next hop. Experiments were conducted under the following scenarios:

1. Multi-Layer Perceptron NN with Back Propagation Learning (MLPNN-BP)
2. Jordan NN with Back Propagation Through Time (Jordan NN-BPTT)
3. Multi Layer Perceptron NN with Bee Algorithm Learning (MLPNN- BA)
4. Jordan NN with Bee Algorithm Learning (Jordan NN – BA)

5. Multi Layer Perceptron NN with Fish School Search Learning (MLPNN- FSS)
6. Jordan NN with Fish School Search Learning (Jordan NN – FSS)
7. Multi Layer Perceptron NN with Proposed Hybrid (MLPNN- Hybrid)
8. Jordan NN with Proposed Hybrid (Jordan NN – Hybrid)

From figure6 Jordan NN hybrid method improved true positive rate by 2.34% and by 3.15% when compared with Jordan NN-FSS and Jordan NN-BA methods respectively. Similarly MLP NN-Hybrid increased true positive rate by 3.44% and 4.65% when compared with MLP NN-FSS and MLP NN-BA methods respectively. Figure 7 shows the average positive predictive value.

From figure7 Jordan NN hybrid method improved Positive Predictive Value by 8.1% and 3.1% when compared with Jordan NN-BP and Jordan NN-BA methods respectively. Similarly MLP NN-Hybrid increased Positive Predictive Value by 8.68% and 4.6% when compared with MLP NN-BP and MLP NN-BA methods respectively.

From figure 8Jordan NN hybrid method improved f measure by 2.3362% and 3.1403% when compared with Jordan NN-FSS and Jordan NN-BA methods respectively. Similarly MLP NN-Hybrid increased f measure by 3.4284% and 4.6478% when compared with MLP NN-FSS and MLP NN-BA methods respectively. It can be observed that both the neural architecture perform equally well when the weights are optimized. Figure 9 shows the misclassification rate.

From figure 9Jordan NN hybrid method reduced misclassification rate by 17.49% and 22.73% when compared with Jordan NN-FSS and Jordan NN-BA methods respectively. MLP NN-Hybridreducedby 23.56% and 30.49% when compared with MLP NN-FSS and MLP NN-BA methods respectively. Figure 10 shows the convergence characteristics of all the meta heuristic technique. It can be seen that BA converges very at about 210 iteration whereas FSS converges later at about 290th iteration. However even with slow converges FSS reduces the MSE considerably. The propose hybrid algorithm converges very fast compared to the other two techniques.

From figure 10, it is observed that the hybrid method given best fitness than the other two methods. The hybrid method achieved lower fitness of 13.94% and 25.26% than FSS and BA methods respectively.

CONCLUSION

Handoff prediction with minimum latency in wireless networks is considered a requirement for improving service quality provided to mobile wireless users. The recent years witnessed much work on mobility prediction schemes development. This paper presents a Jordan network optimized with a hybrid algorithm for the prediction of user movements. One month trace data of Dartmouth College is used for evaluation. Experimental results show that the proposed Jordan network and the proposed hybrid algorithm perform satisfactorily.

REFERENCES

1. Kivi, "Mobile Data Service Usage Measurements - Results 2005-2007", *COIN National Project - Helsinki University of Technology*, 2008 Project Report, http://www.netlab.hut.fi/tutkimus/coin/.
2. J. Chan, S. Zhou and A. Seneviratne, "A QoS Adaptive Mobility Prediction Scheme for Wireless Networks", *IEEE GLOBECOM*, November 1998.
3. S. Kwon, H. Park and K. Lee, "A Novel Mobility Prediction Algorithm Based on User Movement History in Wireless Networks", In *Systems Modeling and Simulation: Theory and Applications: Third Asian Simulation Conference, AsianSim* 2004.
4. S. Michaelis and C. Wietfeld, "Comparison of User Mobility Pattern Prediction Algorithms to Increase Handover Trigger Accuracy", *IEEE 63rd Vehicular Technology Conference*, Spring 2006.
5. I. F. Akyildiz and W. Wang, "The Predictive User Mobility Profile Framework for Wireless Multimedia Networks", *IEEE/ACM Transactions on Networking*, 2004; **12**(6).
6. T. Liu, P. Bahl and I. Chlamtac, "Mobility Modeling, Location Tracking, and Trajectory Prediction in Wireless ATM Networks", *IEEE Journal on Selected Areas in Communications*, 1998; **16**(6).
7. S. Lu, R. Srikant and V. Bhaghavan, "Adaptive Resource Reservation for Indoor Wireless LANs", Proceedings of IEEE GLOBECOM'96, London, 1996.
8. W. T. Poon and E. Chan, "Traffic Management in Wireless ATM Networks Using a Hierarchical

Neural-Network Based Prediction Algorithm", Proceedings of 15th International Conference on Computer and their Applications, ICSA 2000, New Orleans, March 2000.

9. Akoush, S., &Sameh, A. (2007, August). Mobile user movement prediction using bayesian learning for neural networks. In Proceedings of the 2007 international conference on Wireless communications and mobile computing (pp. 191-196). ACM.

10. Kaaniche, H., &Kamoun, F. (2010). Mobility prediction in wireless ad hoc networks using neural networks. arXiv preprint arXiv:1004.4610.

11. Soriano, G. C., &Urano, Y. (2011, February). Replication with state using the self-organizing map neural network. In Advanced Communication Technology (ICACT), 2011 13th International Conference on (pp. 383-388). IEEE.

12. Abu-Ghazaleh, H., & Alfa, A. S. Application of mobility prediction in wireless networks using markov renewal theory. Vehicular Technology, *IEEE Transactions on,* 2010; **59**(2), 788-802.

13. Fotouhi, H., Alves, M., Koubaa, A., &Baccour, N. (2010, August). On a reliable handoff procedure for supporting mobility in wireless sensor networks. In the 9th International Workshop on Real-Time Networks RTN'2010 in conjunction with the 22nd Euromicro International Conference on Real-Time Systems (ECRTS 2010), Brussels, Belgium.

14. Wei-Xin Ling, andYun-Xia Wang , "Using Modular Neural Network with Artificial Bee Colony Algorithm for Classification" , Advances in Swarm Intelligence, Springer, **7928**, 2013, pp 396-403.

15. Slami Saadi, Maamar Bettayeb, Abderrezak Guessoum, and M. K. Abdelhafidi, " Artificial Bees Colony Optimized Neural Network Model for ECG Signals Classification " , Neural Information Processing, Springer, 2012, **7666**, pp 339-346.

16. Hong Yu et al., "Transformer fault diagnosis based on improved artificial fish swarm optimization algorithm and BP network ", IEEE International Conference on Industrial Mechatronics and Automation (ICIMA), May 2010.

17. Huawang Shi and Wanqing Li, "Artificial neural networks with ant colony optimization for assessing performance of residential buildings" , IEEE International Conference on Bio-Medical Information Engineering, December 2009.

18. Mourad, Farah, Chehade, Hicham, Snoussi, Hichem and Yalaoui, Farouk, "Controlled Mobility Sensor Networks for Target Tracking Using Ant Colony Optimization ", *IEEE Transactions on Mobile Computing,* **11**(8).

19. Parija, S,Ranjan, R.K. and Sahu, P.K, "Location prediction of mobility management using neural network techniques in cellular network", International Conference on Emerging Trends in VLSI, Embedded System, Nano Electronics and Telecommunication System (ICEVENT), 2013.

20. http://crawdad.cs.dartmouth.edu

21. P. Stagge and B. Senho. . An extended elman net for modelling time series. In International Conference on Artificial Neural Networks, 1997.

22. Tomasz J. Cholewo and Jacek M. Zurada. Neural network tools for stellar light prediction. In IEEE Aerospace Conference,1997.

23. V.M. Mladenov A. C. Tsakoumis, S. S. Vladov. Electric load forecasting with multilayer percetron and elman neural network. In IEEE NEUREL, 2002.

24. S. Lawrence C. L. Giles and AH C. Tsoi. Noisy time series prediction using recurrent neural networks and grammatical inference. Machine Learning,, **43**:161183, 2001.

25. C. Sitte and J. Sitte. Neural networks approach to the randomwalk dilemma of nancial time series. *Applied Intelligence,* 2002; **16**:163171.

26. M.I. Jordan. Attractor dynamics and parallelism in a connectionist sequential machine. In Proc. of the Eighth Annual Conference of the Cognitive Science Society, pages 531-546. NJ: Erlbaum, 1986.

27. M.I. Jordan. Serial order: A parallel distributed processing approach. Technical report, Institute for Cognitive Science. University of California, 1986.

28. R. C. Eberhart, and Y Shi. Computational Intelligence: Concepts to Implementations, Morgan Kaufmann Publishers. San Francisco, (in press).

29. Marco Gori, Alberto Tesi, On the problem of local minima in back-propagation, *IEEE Trans. Pattern Anal. Mach. Intell,* 1992; **14**(1): 76–86.

30. A. van Ooyen, B. Nienhuis, Improving the convergence of the back-propagation algorithm, *Neural Network,* 1992; **5**(4): 465–471.

31. M. Ahmad, F.M.A. Salam, supervised learning using the Cauchy energy function, in: Proc. of International Conference ON Fuzzy logic and Neural Networks, 1992.

32. R.A. Jacobs, Increased rates of convergence through learning rate adaptation, *Neural Networks* 1988; **1;** 295–307.

33. M.K. Weirs, A method for self-determination of adaptive learning rates in back propagation, *Neural Networks* 1991: **4;** 371–379.

34. Karaboga, N., A. Kalinli, and D. Karaboga, \Designing digital IIR filters using ant colony optimisation algorithm," *Engineering Applications of Artificial Intelligence*, 2004; (3): 301{309.
35. Karaboga, D. and B. Basturk, \On the performance of artificial bee colony (ABC) algorithm," Applied Soft Computing, Vol. 8, No. 1, 687{697, 2008.
36. Von Frisch K. Bees: Their Vision, Chemical Senses and Language. (Revised edn) Cornell University Press, N.Y., Ithaca, 1976.
37. Seeley TD. The Wisdom of the Hive: The Social Physiology of Honey Bee Colonies. Massachusetts: Harvard University Press, Cambridge, 1996.
38. Bonabeau E, Dorigo M, and Theraulaz G. Swarm Intelligence: from Natural to Artificial Systems. Oxford University Press, New York, 1999.
39. Camazine S, Deneubourg J, Franks NR, Sneyd J, Theraula G and Bonabeau E. Self-Organization in Biological Systems. Princeton: Princeton University Press, 2003.
40. Bastos-Filho, C.J.A., Neto, F.B.L., Lins, A.J.C.C., Nascimento, A.I.S., Lima, M.P.: A novel search algorithm based on fish school behavior. In: IEEE International Conference on Systems, Man and Cybernetics, pp. 2646–2651. IEEE, Los Alamitos (October 2009)
41. Bastos-Filho, C.J.A., Neto, F.B.L., Sousa, M.F.C., Pontes, M.R.: On the Influence of the Swimming Operators in the Fish School Search Algorithm. In: SMC, pp. 5012–5017 (October 2009)