Prediction-based location management using multilayer neural networks

B. P. Vijay Kumar and P. Venkataram*
Protocol Engineering and Technology (PET) Unit, Department of Electrical Communication Engineering,
Indian Institute of Science, Bangalore 560 012, India.
email: {vijaybp,pallapa}@ece.iisc.ernet.in

Abstract

Location management is one of the key issues in mobile networks to provide an efficient and low-cost services. In this paper, we propose a prediction-based location management scheme for locating a mobile host (MH), which depends on its history of movement pattern. A multilayer neural network (MNN) model for mobile movement prediction is designed to predict the future movement of a mobile host. The MNNs are trained with respect to the data obtained from the movement pattern of a mobile host for making predictions.

The performance of the method has been verified for prediction accuracy by considering different movement patterns of a mobile host. The simulation has achieved an average of 93% prediction accuracy for uniform movement, 40% to 70% for regular movement and 2% to 30% for random movement patterns of an MH.

Keywords: Location management, movement prediction, multilayer neural network.

1. Introduction

Future mobile networks will have to support large population of mobile users and have to provide efficient and low-cost services under diversified characteristics of network architecture, services, and user types, i.e. different cell sizes, multimedia services, different mobility users, etc. One of the most important issues in the mobile networks is the location management.

Location management is the process by which the current location of a mobile host (MH) is determined. It can be divided into two different services: *mobile tracking* and *locating*. *Mobile tracking* is the process by which the network keeps track of the current location of the mobile host, whereas *Mobile locating* deals with the process of finding the current location of the mobile host for the delivery of an incoming call.

One way of searching an MH to identify its current location is to broadcast a search information, called *paging*, in each and every cell in the whole geographical area [1]. However, the amount of channel bandwidth consumed by these numerous broadcast signals can be extremely high. The other way of searching is to store the current location of each mobile host by a *location update* signal from the mobile hosts and maintaining the location information database for each mobile host [2], which is more expensive and has high signalling traffic.

*Author for correspondence.

In mobile networks, location updating and paging generate enormous traffic. Several attempts have been made in this connection [1]–[3], which need to be reduced. One of the methods is to use mobile host movement behaviour and their traffic characteristics to predict the future location of a mobile host. If the location of an MH is known in advance, then no explicit location updation is necessary and paging throughout the geographical area can be avoided. Thus, by maintaining the knowledge of the MH's history of movement behaviour, the signalling overload can be reduced.

In this paper, we propose a prediction-based location management using multilayer neural network (MNN). The method predicts the future location of a mobile host based on the history of its movement pattern. The MNN is trained with respect to the data obtained from the history of movement pattern of a mobile host for making predictions.

This paper is organized as follows. In Section 2, the related works are given. Some of the definitions, neural network training procedure and MNN model for mobile movement predictions are described in Section 3. A prediction-based location management is illustrated in Section 4. Simulation and results are discussed in Section 5, and finally Conclusions are given in Section 6.

2. Related work

Most of the existing works on location management consider mobile host's movement information, present location and velocity for tracing the mobile host. We classify some of the works carried out for location management into two groups: firstly, the works on location management based on mobile movement behavior and later prediction-based location management are discussed.

The method proposed in [4] aims to reduce signalling load resulting from location tracking. The idea is to take recent user movement information (called the paging information record) into account to determine which location area to page first. An effective location management strategy based on user mobility classes is tested and compared in [5] along with intelligent paging to reduce cost and paging delay. It shows that location management based on user mobility performs better. An optimal location registration area based on various mobility patterns of users and network architecture is designed to minimize the rate of location update, which in turn reduces the cost of tracking the mobile host [6]. The user profile (indicating the mobile host's, identity and current location), replication mechanism for faster location lookup of a mobile user in a personal communication system (PCS) is presented with a minimum-cost maximum-flow-based algorithm to compute the set of sites at which a user profile should be replicated for given known calling and user *mobility patterns* [7]. A history-based location update scheme is considered [8] to reduce the number of unnecessary location updates by taking the recent mobility history of the mobile terminal into account to show that it results in a lower cost for mobility management. In [9], dynamic location areas are determined for each user on the basis of gathered statistics and incoming call patterns to reduce the average signalling cost to track a mobile user. Also, an optimal multistep paging algorithm is used to minimize the paging signalling cost.

Most of the works on predictive location management are based on the mobile host's movement history, i.e. previous movement patterns. We discuss some of them here.

An efficient heuristics to predict the location of a mobile user in a cellular network is presented in [10]. It assumes a hierarchy of location areas, which changes dynamically with traffic patterns. Depending on the movement profile of a user, it is possible to compute most probable location area and a future probable location area for the user. A mobile tracking scheme that exploits the predictability of user mobility patterns in wireless PCS networks is presented in [11]. In this scheme, a Gauss–Markov model is used and mobile's future location is predicted based on the information gathered from the mobile's last report of location and velocity. In [12], several basic prediction algorithms using real-life movement traces are verified, and a QoS adaptive mobility prediction is introduced to resolve the limitations of an individual's movement history for mobility prediction. The work carried.out by introducing a dynamic velocity paging scheme based on semi-real-time movement information of an individual user, which allows a more accurate prediction of the user location at the time of paging, is reported in [13]. A novel predictive mobility management algorithm for supporting global mobile data accessing is introduced in [14]. This algorithm predicts the future location of a mobile user according to the user's movement history, i.e. previous movement patterns.

3. Proposed method for mobile movement prediction

The proposed method for mobile movement prediction (MMP) is based on the mobile host history of movement patterns, which has been recorded for a certain time duration. Multilayer neural networks are used to process the mobile movement pattern for accurate prediction of mobile movements. Before discussing the method, we present some of the definitions used in this paper and the neural network training procedure which has been implemented for MMP.

3.1. Definitions

Definition 1: *Movement pattern* (P_n) is the history of movement of a mobile host recorded for a period of time T_n, where n is the number of regular time intervals at which the mobile host movements are recorded. The time interval can be minutes, hours, days, etc. The movement pattern P_n can be represented by a data at regular time interval, $t_1, t_2, ...t_n$.

Let the movement pattern $P_n = \{p_1, p_2, ...,p_n\}$ be recorded for a mobile MH, where P_i indicates the movement of a mobile host during time t_i, and we define the movement in terms of direction and distance travelled by an MH during the time interval t_i. Then P_i is represented by a pair (d_i, ds_i).

- d_i is the possible direction in which a mobile host moves at i^{th} time interval. For example, if we consider four possible directions, *North, East, South* and *West*, of movement of a mobile host, then $d_i \in \{North, East, South, West\}$. If a mobile host moves towards North direction at i^{th} time interval, then $d_i = North$.

- ds_i is the distance travelled by a mobile host at i^{th} time interval. Here, the distance travelled may be number of cells, kilometers, meters, etc.

For example, if a mobile movement pattern is recorded for three time intervals $(n=3)$ with direction of movements, North, East, East and the distance travelled is 2, 1, and 3 units, then the movement pattern is, $P_3 = \{p_1, p_2, p_3\} = \{(d_1, ds_1), (d_2, ds_2), (d_3, ds_3)\} = \{(North, 2), (East, 1), (East, 3)\}$.

Table I

Training data set

Subpattern	$Input_1$	$Input_2$	$Input_3$	$Input_4$	Desired Output
1	p_1	p_2	p_3	p_4	p_5
2	p_2	p_3	p_4	p_5	p_6
3	p_3	p_4	p_5	p_6	p_7
4	p_4	p_5	p_6	p_7	p_8
5	p_5	p_6	p_7	p_8	p_9
6	p_6	p_7	p_8	p_9	p_{10}

Definition 2: *Training data set* is the set of subpatterns obtained from the movement pattern p_n by partitioning it into $n - k$ subpatterns, where $k + 1$ is the size of each subpattern ($k \ll n$). The subpattern is a training data pair with mobile movements for k time intervals as input and the movement for the next time interval as a desired output. For example, the first training subpattern is $p_1, p_2, ..p_k$ as input and p_{k+1} as the desired output.

The parameter k is the prediction order or time window, which is chosen based on the movement characteristics of a mobile host and the size of the recorded movement pattern [15], [16].

Table I shows the training data set obtained from the movement pattern of p_{10} and $k = 4$.

In general, suppose we have a movement pattern for n time intervals (time-lagged intervals), then for prediction order of k, there are $n - k$ training *subpatterns*. The first training subpattern is $p_1, p_2, ..p_k$ as the inputs and p_{k+1} as the desired output. Similarly, the ith training subpattern contains $p_i, p_{i+1}, ..., p_{i+k-1}$ as the inputs and p_{i+k} as the desired output.

3.2. Design and training procedure for neural network

The multilayer neural network devised for mobile movement prediction is based on the back-propagation learning algorithm [17], [18]. Here, the neural networks are trained with respect to movement pattern for learning optimized functions for predictions [19]. The role of the neural networks in this application is to capture the unknown relation between the past and the future values of the movement pattern. This helps in predicting the future location of a mobile host for location management.

3.2.1. Selection of neurons for multilayer neural network model

This is the most important factor to be considered to develop an MNN model of appropriate size for capturing the underlying movement patterns in the training data. The selection of number of input and hidden layer neurons will largely fix the size of the MNN model. There are some thumb rules proposed, whose guidelines are some heuristic methods.

The neural network model we consider for prediction consists of three layers: input, hidden and output. The number of input neurons is an important parameter since it corresponds to the length of the subpatterns used to discover the underlying features in a given movement data. Some reports suggest that too few or too many input neurons can have significant impact on the

learning and prediction ability of the network [15]. In practice, the number of neurons is often chosen through experimentation or by trial and error to have more generalization capability for the MNN model. Also, while choosing the number of input and hidden layer neurons, care must be taken to avoid any underlearning or overfitting of the training data.

A set of input layer neurons is selected by experimentation results and correspondingly the length of the input training subpatterns or window size k of the training data that is considered for a given movement pattern.

For example, Table I shows that the number of movements considered in the input training data is 4, i.e. $k = 4$, and each movement p_i requires two neurons to represent direction (d_i) and the distance (ds_i) travelled by an MH. Hence, the number of input layer neurons for the above representation is equal to $k*2 = 8$.

The number of neurons in the hidden layer depends on the length of the subpattern and the number of subpatterns provided for training [16], [18]. The number of output layer neurons depends on the output movement parameters and their representation. In our case, we consider direction and distance as the movement parameter; hence, there are two output neurons.

3.2.2. Training procedure

We explain the training procedure for a general MNN model by considering L layers with M input neurons in the MNN model by using backpropagation learning algorithm. Each layer of MNN consists of more than one neuron. The output of lth layer is fed as the input of the $(l+1)^{th}$ layer and the neurons in l^{th} layer are connected to those in the $(l+1)^{th}$ layer with an adaptive weight (see Fig. 1).

Suppose the total input to the jth neuron in the first hidden layer (h) (i.e. 2nd layer) is T_j, then for M neurons in the input layer,

$$T_j = \sum_{i=1}^{M} \left(w_{ji}^h * output_i \right) \tag{1}$$

where w_{ji} is the weight of the link connecting neuron i in the input layer to the neuron j in the hidden layer and $output_i$ is the output from the neuron i in the input layer. The output of each neuron is determined by applying a transfer function $f(.)$ to the total input to the neuron. We use the sigmoid activation function

$$f(x) = (1 + e^{-x})^{-1}. \tag{2}$$

The training is done by using backpropagation in two passes. The forward pass is used to evaluate the output of the neural network for the given input in the existing weights. In the reverse pass, the difference in the neural network output with the desired output is compared and fed back to the neural network as an error to change the weights of the neural network (see Fig. 2).

In the reverse pass, suppose for the particular neuron i in the output layer L, the output is o_i^L while the desired output value is d_i^L. The backpropagated error θ_i^L is:
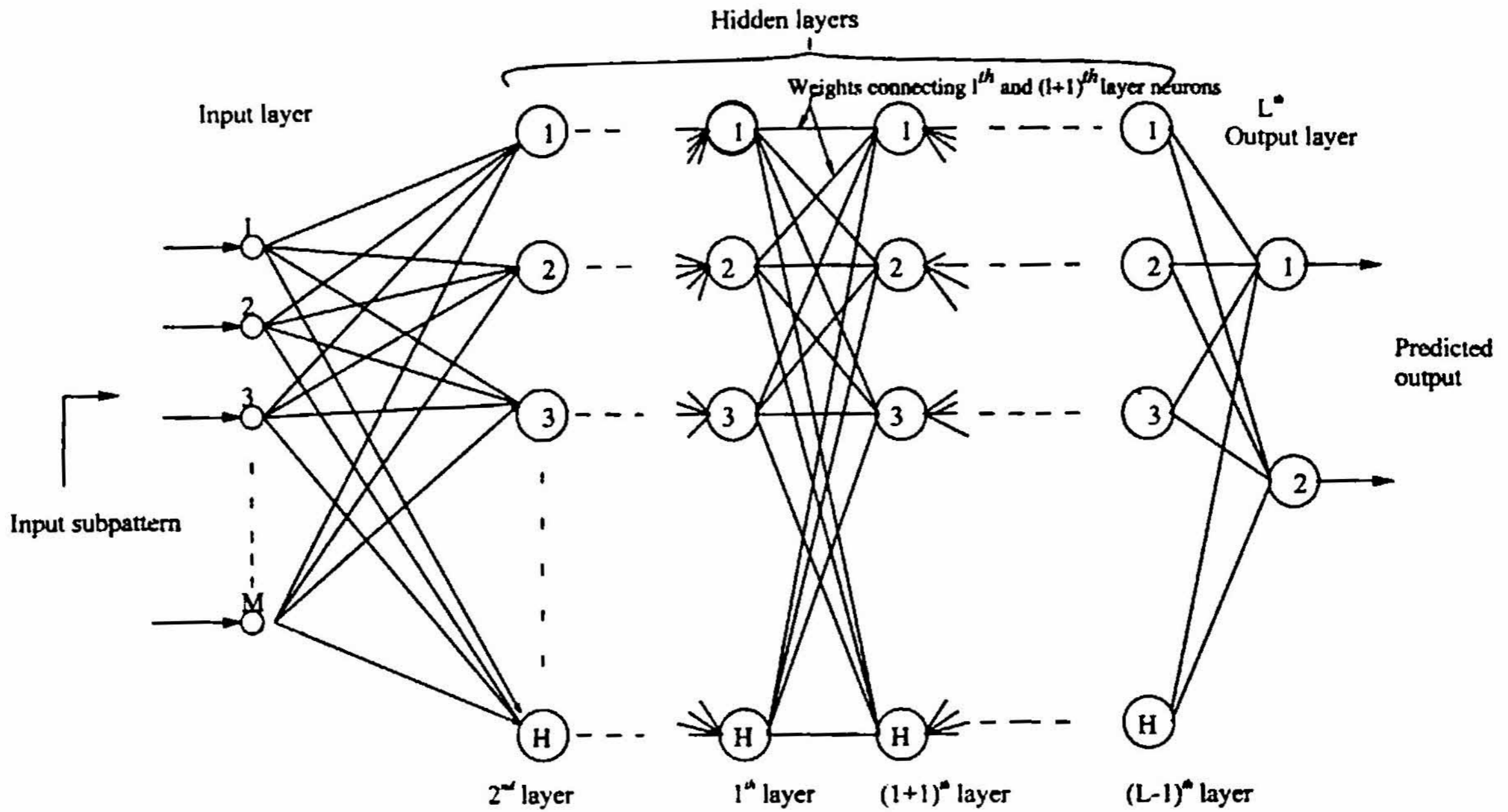
Fig.1. Multilayer neural network architecture.

$$\theta_i^L = f'(T_i^L)[d_i^L - o_i^L] \tag{3}$$

where T_i^L represents the total input to the ith neuron in the output (L^{th}) *layer* and $f'(.)$ is the derivative of $f(.)$.

For layers $l = 1, 2, .., (L-1)$ and for each neuron i in layer l the error is computed using

$$\theta_i^l = f'(T_i^l) \sum_j (w_{ij}^{l+1} * \theta_j^{l+1}). \tag{4}$$

The adaptation of the weight of the link connecting neurons i of layer l and neuron j of layer $(l-1)$ is done by using

$$w_{ij}^l = w_{ij}^l + \alpha * \theta_i^l * o_j^{l-1} \tag{5}$$

where α is the learning constant that depends on the rate at which the neural network is expected to converge. This neural network algorithm essentially minimizes the mean square error between the MNN output and the desired output using gradient descent approach [17].

Figure 2 illustrates the neural network training model for mobile movement predictions. The actual predicted output $o = \hat{p}_{i+k}$ is compared to the desired output $d = p_{i+k}$ and the error values
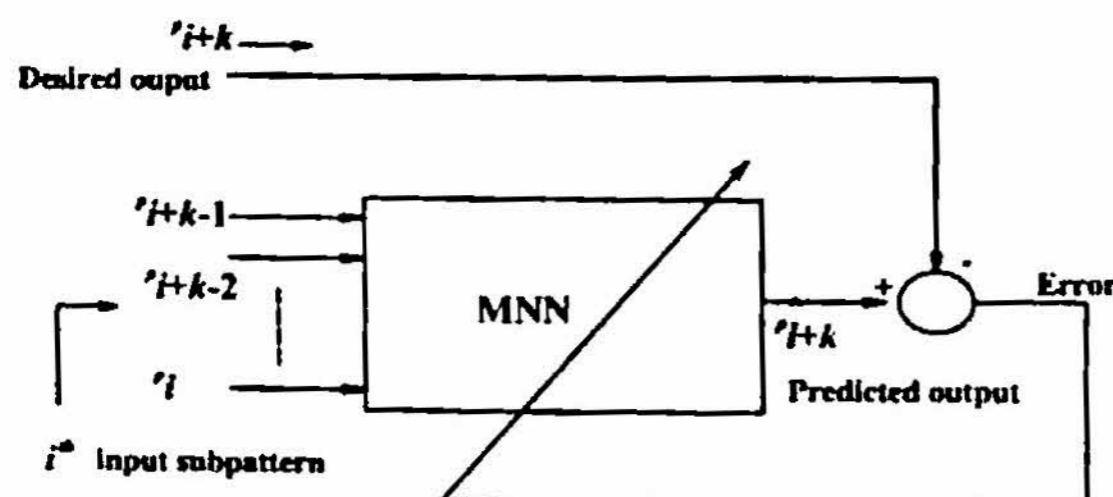

Fig. 2. Neural network training model for mobile movement prediction.

are used to calculate new weights of connections between neurons of all input, output and hidden layers using eqn (5), thereby reducing the error in the output.

This training procedure is iterated over all the entries of the training data set for several times until the mean square error ($\sum [d_i^L - o_i^L]^2$) between neural network output and the desired output reaches some specified threshold, which for example, could be 0.005.

3.3. MNN model for mobile movement prediction

The mobile movement prediction is to find the future movement of a mobile host from the MNN model trained with respect to the training data set. To predict the future movement of a mobile host, we can either go for single or multiple move prediction.

3.3.1. Single move prediction

This predicts the movement (p_{n+1}) of a mobile host at time interval t_{n+1} for the given movement pattern p_n. To carry out single move prediction, input the subpattern $\{p_{n-k+1}, p_{n-k+2}, ... p_n\}$ to the MNN and the output obtained will be the predicted movement p_{n+1} for time interval t_{n+1}, which is the direction d_{n+1} and the distance ds_{n+1} travelled by a mobile host. Figure 3 illustrates the single move prediction model.

3.3.2. Multiple move prediction

This predicts the movement of a mobile host after several time intervals from t_n^{th} time interval, i.e. to predict the movement (p_{n+m}) of a mobile host, where $m > 1$. A recursive method has been designed for multiple move predictions. In this method, the predicted output p_{n+1} is inserted as one of the inputs in subpattern at the extreme right by shifting the entire subpattern to left by one time interval (see Table 1) to predict the next movement p_{n+2} as shown in Fig. 4.

For example, to obtain p_{n+m} predicted output, the first subpattern input to MNN model is $\{p_{n-k+1}, p_{n-k+2}, ...p_n\}$ and p_{n+1} is the predicted output. The second subpattern contains $\{p_{n-k+2}, p_{n-k+3}, ...p_{n+1}\}$ as the input and p_{n+2} as the predicted output. Finally, the last subpattern is $\{p_{n+m-k}, p_{n+m-k+1}, ...p_{n+m-1}\}$ and p_{n+m} as the predicted output.

4. Prediction-based location management

This section discusses the prediction of future location of a mobile host by using MNN model for mobile movement prediction.

The movement pattern of an MH travelled over a period of time $[0, T_n)$ is recorded and is processed to construct MNN model for mobile movement prediction. If an MH is to be located (or called) at time T_c with $T_c > T_n$, the system calculates the time difference between T_n and T_c to
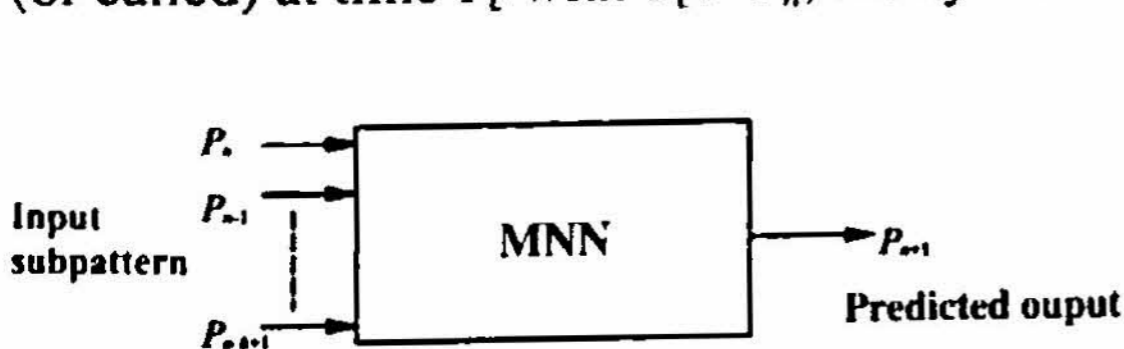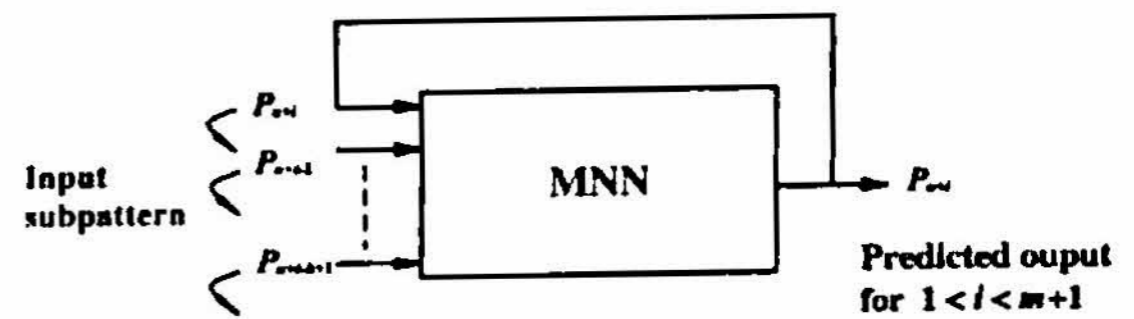

Fig. 3. MNN model for single move prediction.

Fig. 4. MNN model for multiple move prediction.

find how many time intervals ahead the prediction is to be carried out, i.e. $\lceil m \rceil = \frac{(T_c - T_n)}{\Delta t}$ time intervals, where $\Delta t = t_i - t_{i-1}$. If $m = 1$, then a single move prediction is to be carried out. If $m > 1$, then a multiple move prediction is carried out. The process of prediction to find MH location is explained with the following example.

For simplicity, without loss of generality, we assume that the cells are rectangular shaped. In our example, we consider an 8 × 8 rectangular array of cells, constituting the mobile network topology (Fig. 5). The mobile movements are recorded in terms of cell number at every time interval. The recorded mobile movements are preprocessed to obtain the movement pattern, i.e. direction of movements and distance travelled, for training the neural network. For example, let an MH move in direction x at i^{th} time interval from the current position, then $d_i = x$. Consider $x \in D = \{North, NorthEast, East, SouthEast, South, SouthWest, West, NorthWest\} = \{N, NE, E, SE, S, SW, W, NW\}$, a set of possible directions, and if an MH is moving at a certain speed, then the distance travelled at i^{th} time interval is $ds_i = y$, where y is the number of cells crossed by an MH during the ith time interval. The movement pattern P_n for a mobile host is obtained as follows.

To determine the direction and distances moved by an MH we use an adjacency matrix corresponding to the neighbours of a cell.

Figure 6 shows the adjacency matrix for the network topology given in Fig. 5, which is a 64 × 8 matrix, where the elements of ith row indicate the neighbours of ith cell in the order of directions specified in the direction set D (neighbours for boundary cells are chosen from the set of boundary cells in accordance with the direction). For space conservation, the first 20 rows of the matrix are shown, i.e. cells 1 to 20.

The rationality behind the prediction of location of a mobile host is based on the following. In general, the movement of mobile hosts shows some pattern according to their movement behaviour. For example, users living in urban areas move daily from home to office and back home. If the movement patterns for the previous several hours, days or months are investigated, some periodicity for the patterns will be exhibited [14]. This periodicity is a key to predicting the future location of a mobile host. Based on the above discussion, the movement patterns of a mobile host may be subdivided into *uniform, regular, random, zigzag* movements, etc. Uniform movements are the ones in which the movement of a mobile host will be in the same direction over a period of time considered. For regular movements the movement pattern will be periodic and deterministic in nature. These patterns will change dynamically if a mobile changes its regular daily movement, e.g. every Saturday is a group discussion, etc. But majority of mobile users have some regular daily movement patterns and follow these patterns more or less every day of the week except during weekends or holidays when the movement is random; hence the movement patterns are stochastic in nature.

To illustrate these, consider three different movement patterns, which are recorded for MHs: M1, M2 and M3. The movement patterns are uniform, regular and random for MHs M1, M2 and M3, respectively, as shown in Fig. 5. Their movement patterns are given in terms of set of cells the mobile has crossed.
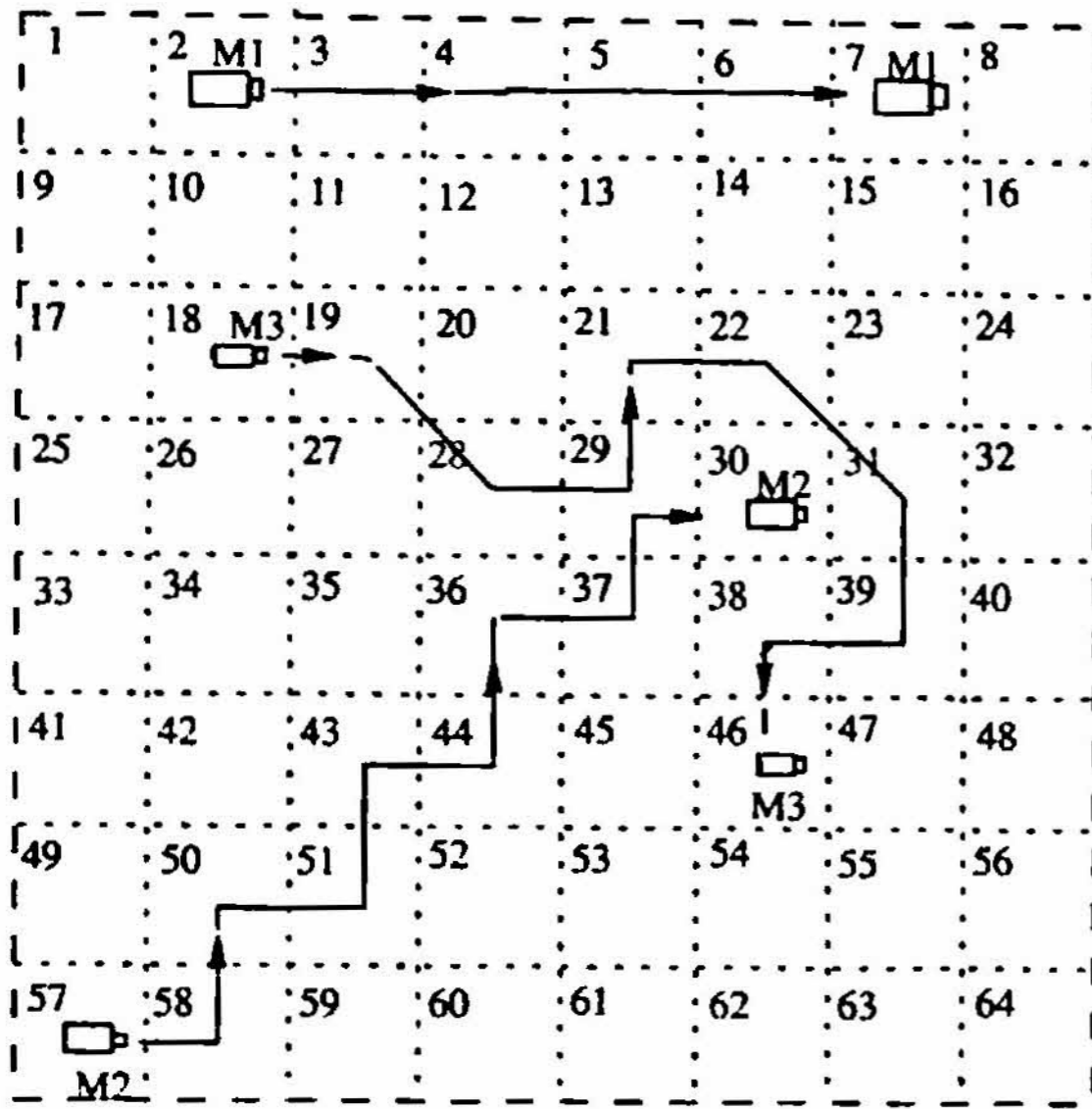
Let the movement patterns be:

Adjacency Matrix

Direction ⟶

Cell #	N	NE	E	SE	S	SW	W	NW
1	57	58	02	10	09	16	08	64
2	58	59	03	11	10	09	01	57
3	59	60	04	12	11	10	02	58
4	60	61	05	13	12	11	03	59
5	61	62	06	14	13	12	04	60
6	62	63	07	15	14	13	05	61
7	63	64	08	16	15	14	06	62
8	64	57	01	09	16	15	07	63
9	01	02	10	18	17	24	16	08
10	02	03	11	19	18	17	09	01
11	03	04	12	20	21	18	10	02
12	04	05	13	21	22	19	11	03
13	05	06	14	22	23	20	12	04
14	06	07	15	23	22	21	13	05
15	07	08	16	24	23	22	14	06
16	08	01	09	17	24	23	15	07
17	09	10	18	26	25	32	24	16
18	10	11	19	27	26	25	17	09
19	11	12	20	28	27	26	18	10
20	12	13	21	29	28	27	19	11
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Fig. 5. An 8 x 8 rectangular array of cells representing the mobile network topology.

Fig. 6. Adjacency matrix for the network topology shown in Fig. 5.

for M1 {2, 3, 5, 6, 7} for 4 time intervals;

for M2 {57, 58, 50, 51 43, 44, 36, 37, 29, 30} for 9 time intervals;

for M3 {18, 19, 28, 29, 21, 22, 31, 39, 38, 46} for 9 time intervals.

From the cell-based movement pattern, we derive the direction and distance by using the adjacency matrix as preprocessing for prediction.

The above movement patterns, preprocessed for obtaining the distance and direction of movement pattern using adjacency matrix (Fig. 6), are given here.

For M1, the movement pattern is

$P_4^1 = \{(E,1), (E,2), (E,1), (E,1)\}$, where P_4^1 is the movement pattern of M1 for four time intervals.

For M2, the movement pattern is

$P_9^2 = \{(E,1), (N,1), (E,1), (N,1), (E,1), (N,1), (E,1), (N,1), (E,1)\}$, where P_9^2 is the movement pattern of M2 for nine time intervals.

For M3, the movement pattern is

$P_9^3 = \{(E,1), (SE,1), (E,1), (N,1), (E,1), (SE,1), (S,1), (W,1), (S,1)\}$, where P_9^3 is the movement pattern of M3 for nine time intervals.

Table II
Training data set for mobile host M3

d_1, ds_1	d_2, ds_2	d_3, ds_3	d_4, ds_4	Output: d_5, ds_5
(E, 1)	(SE, 1)	(E, 1)	(N, 1)	(E, 1)
(SE, 1)	(E, 1)	(N, 1)	(E, 1)	(SE, 1)
(E, 1)	(N, 1)	(E, 1)	(SE, 1)	(S, 1)
(N, 1)	(E, 1)	(SE, 1)	(S, 1)	(W, 1)
(E, 1)	(SE, 1)	(S, 1)	(W, 1)	(S, 1)
(SE, 1)	(S, 1)	(W, 1)	(S, 1)	X

By observing the directional changes in the movement pattern, suitable prediction order (k) is considered and the corresponding subpatterns are obtained. These subpatterns are used for training MNN.

The training data set for mobile host M3 with movement pattern P_9^3 for $k = 4$ in the above example is shown in Table II where

d_1 and ds_1 = Mobile host direction and distance during the first time interval, respectively;

d_2 and ds_2 = Mobile host direction and distance during the second time interval; and so on.

d_5 and ds_5 = Mobile host direction and distance observed at time interval 5, i.e. desired output for the given input training data.

The neural network is trained with all the subpatterns to predict the movement X of Table II. This illustrates the single move prediction for MH3. The same data set can be used to predict multiple moves of MH3 by updating the data set given in Table II by multiple move prediction algorithm.

Algorithm: Prediction-based location management

Nomenclature

P_n = Recorded movement pattern for time duration $[0, T_n)$;

T_n = Time, up to which the movement pattern is recorded;

T_c = Time at which the location of a mobile host is to be predicted;

Δt = Time interval, at which mobile movements are recorded;

m = Number of time intervals ahead of the prediction of location of a mobile host;

BEGIN

Step 1: Movement pattern P_n for an MH is recorded for a time duration $[0, T_n)$;

Step 2: Derive the direction and distance from the movement pattern using adjacency matrix;

Step 3: *Subpatterns* for the movement pattern are derived as a training data set for training MNN;

Step 4: MNN is trained as per the training procedure in Section 3.2, using training data set;

Step 5: IF MH is to be located at time $T_c > T_n$ THEN

Compute $[m] = \frac{(T_c - T_n)}{\Delta t}$

IF $(m > 1)$ THEN

Carry out *multiple move prediction* to predict the movement p_{n+m};

ELSE

IF $(m = 1)$ THEN

Carry out *single move prediction* to predict the movement p_{n+1};

Step 6: Using adjacency matrix, find the location of an MH from the predicted direction and distance, given the current location of an MH;

END

Here the training of neural network is carried out once for a given movement pattern of an MH, and is retrained if the prediction accuracy decreases by certain threshold level. This decrease in prediction accuracy is mainly due to change in daily regular movement of an MH. These deviations in the movement pattern of a mobile host are used for retraining of the MNN. These retraining processes take very few iterations to learn the movement behaviour. Hence the computational complexity depends on how often the neural networks are retrained due to change in daily regular movement of an MH.

5. Simulation

The simulation is carried out for various sizes of mobile cell environments. To describe the simulation we considered the mobile environment which contains an 8 x 8 rectangular array of cells in which the mobile hosts can move freely. The possible direction which a mobile host can move is considered from the direction set $D = \{North, NorthEast, East, SouthEast, South, SouthWest, West, NorthWest\} = \{N, NE, E, SE, S, SW, W, NW\}$. The distance is recorded in terms of the number of cells travelled by a mobile host at each time interval.

The MNN model is devised with k neurons in the input layer, where k is made to vary depending on the mobile movement pattern for better prediction accuracy. Numerous trial runs indicated that the accuracy of prediction is not sensitive to small changes in the number of input neurons. We also observed that too few or too many input neurons can have significant impact on the learning and prediction ability of the network, with too few input neurons resulting in underlearning and too many resulting in overfitting (learning). In the simulation, we chose 4, 6 and 8 input neurons for uniform, regular and random movement patterns, respectively, and have obtained a

better prediction performance. Also, we chose a number of hidden layer neurons based in the experimentation to avoid under or overfitting. We chose 20 neurons in hidden layer and two in the output layer. The model is developed in C++ programming language on a Pentium III work station. All training data are normalized into real values between 0.0 and 1.0, since every neuron in the hidden and output layer of the network employs a *sigmoid* function. We chose the value of learning parameter $\alpha = 0.1$.

For a mobile host, we have considered a desired movement pattern recorded over time intervals of 100 to 200. The training of MNN is performed by using the first 50% of the desired movement pattern as training data set and the remaining as a test data set for predictions. Also, the training is performed by picking a portion of the movement pattern in random as training data set and the prediction test is carried out over the remaining portion of the movement pattern.

An example of the pattern with training and test data set are shown in Fig. 7. The results are taken for both single and multiple move predictions.

Simulation is carried out by considering different sets for each movement patterns like uniform, regular, zigzag and random, etc. We also have experimented by training MNN for change in movement values of a given pattern which has taken lesser number of iterations to achieve the same accuracy and learning error. This helps in adapting MNN for any change in the daily movement of a mobile host for prediction. The results are taken for average learning error of MNN and average prediction accuracy for single and multiple move predictions.
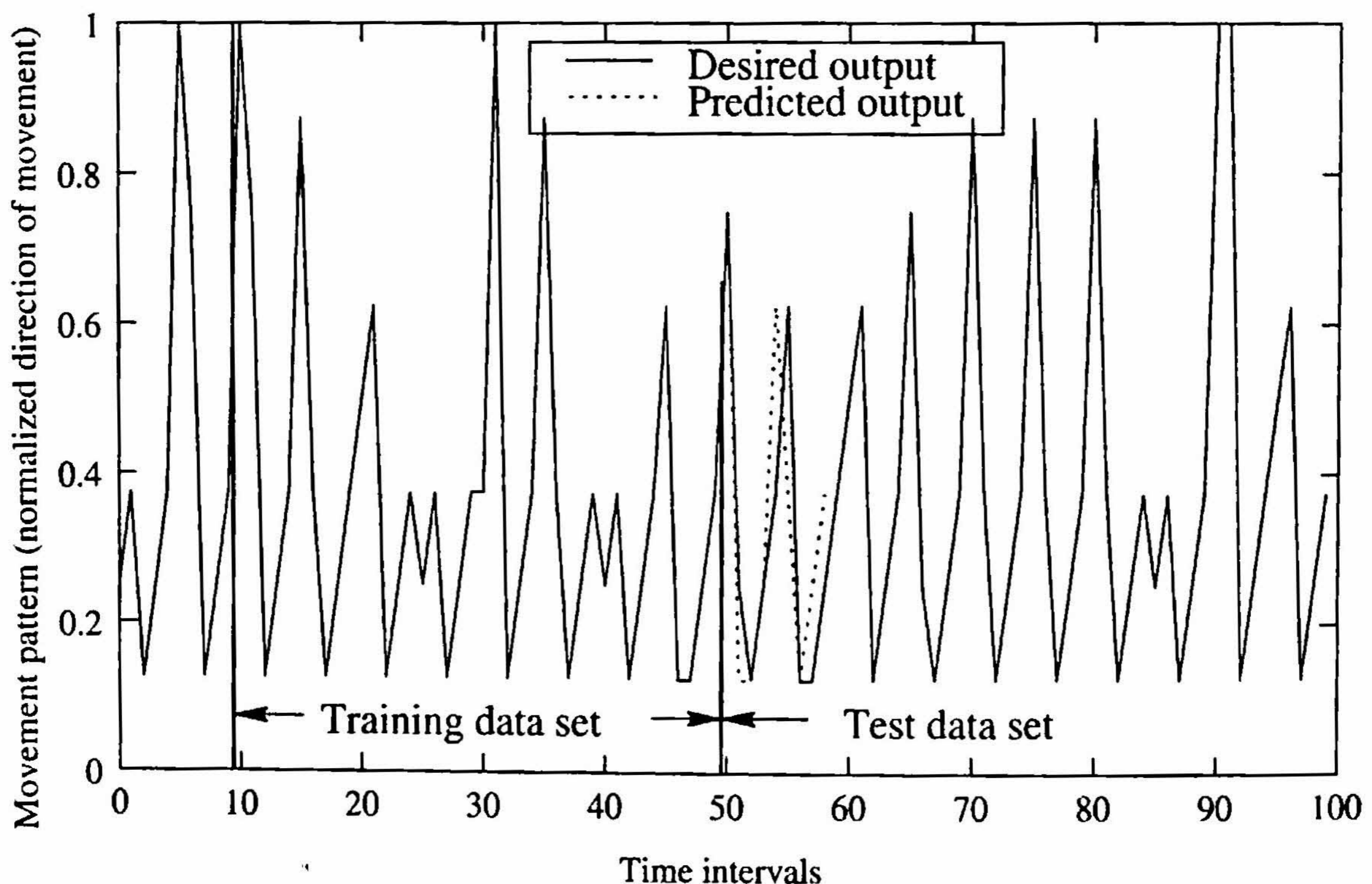

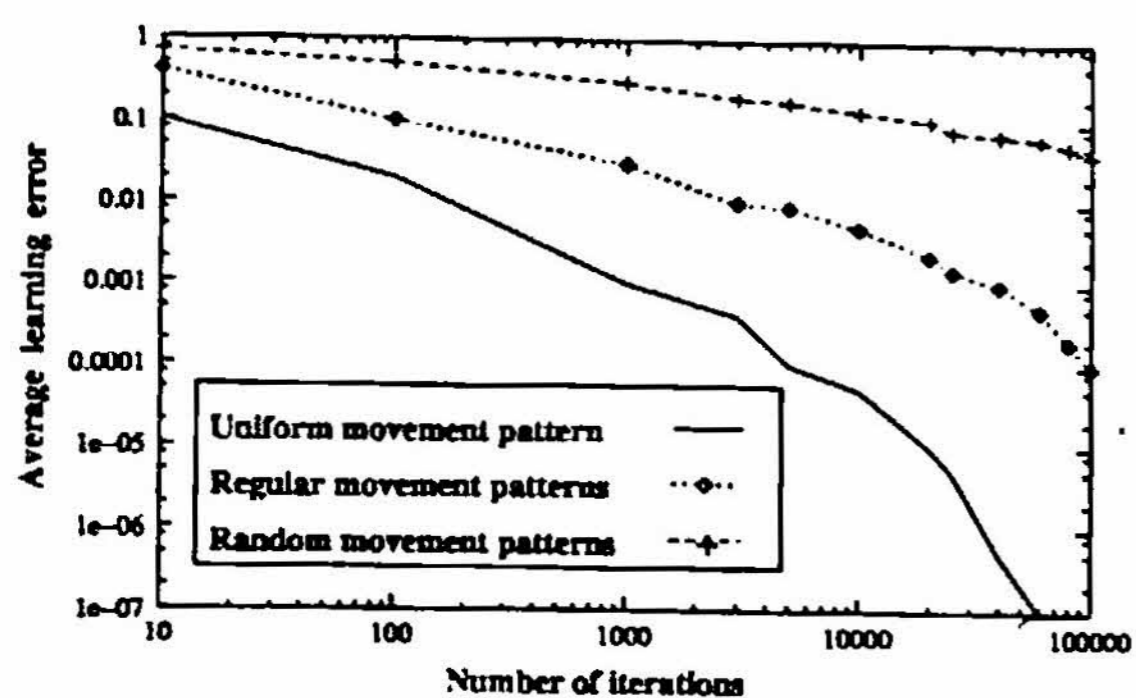Fig. 7. Comparison of predicted movement versus desired movement output.

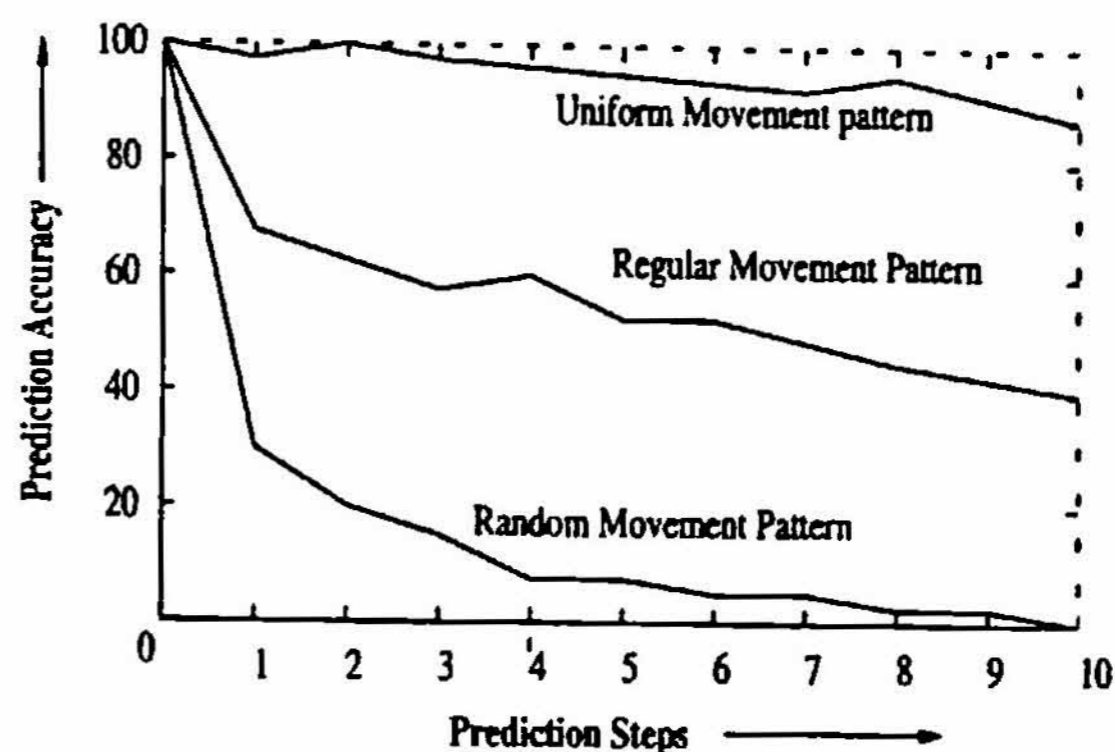Fig. 8. Learning error with respect to number of iterations.

Fig. 9. Average prediction accuracy of MNN model.

We measure the average learning error and prediction accuracy as follows.

Average learning error $= \frac{1}{100-k} \sum_{i=k}^{100}(p_i - p'_i)^2$, where p_i denotes the desired output, p'_i the prediction output at i^{th} time interval and k the prediction order.

$$\text{Prediction accuracy} = \frac{Number\ of\ times\ the\ correct\ prediction\ of\ location\ of\ an\ MH}{Total\ number\ of\ times\ the\ prediction\ of\ location\ of\ an\ MH}.$$

Results obtained by considering different sets for each movement pattern to find an average prediction accuracy are presented.

5.1. Results

The results obtained for average learning error and prediction accuracy for different movement patterns (i.e. uniform, regular and random) are shown in Figs 8 and 9. The graph shown in Fig. 8 is plotted for learning error versus number of iterations carried out during the training of MNN.

From the graph, it is observed that the number of training iterations required for regular movement patterns is much less than for random patterns, for a given learning error threshold value (= 0.005). These results show the time required for training a neural network for a given movement pattern of an MH.

Figure 9 depicts the graph for an average prediction accuracy of MNN with respect to the number of prediction steps. It is observed that the average prediction accuracy for uniform patterns is 93%, 40% to 70% for regular patterns and decreases drastically for random patterns (varies from 2% to 30%) with respect to the time intervals. Thus, as the prediction steps increase the accuracy decreases and drops drastically, especially for random movement pattern.

6. Conclusion

We propose a prediction-based location management in a mobile network. The approach uses a multilayer neural network to predict the future location of a mobile host based on the history of movement pattern of a mobile host. MNN model for single and multiple move prediction is designed for predicting the future location of a mobile host. The performance of the method has

been verified for prediction accuracy by considering different movement patterns of a mobile host and learning accuracy of the MNN model.

Simulation is also carried out for different movement patterns (i.e. regular, uniform, random) to predict the future location of a mobile host. The average prediction accuracy was measured and achieved up to 93% accuracy for uniform patterns, 40% to 70% for regular patterns and 2% to 30% for random movement patterns. The proposed method helps in reducing the signalling cost for location management by predicting the future location of a mobile host.

References

1. C. Rose and R. Yates, Minimizing the average cost of paging under delay constraints, *Wireless Networks*, 1, 211–219 (1995).

2. C. Rose. Minimizing the average cost of paging and registration: A timer based method. *Wireless Networks*, 2, 109–116 (1996).

3. S. J. Kim and C.Y. Lee, Modeling and analysis of the dynamic location registration and paging in micro-cellular systems. *IEEE Trans.*, VT-45, 82–89 (1996).

4. K. Wang, J.-M. Liao and J.-M. Chen, Intelligent location tracking strategy in PCS, *IEE Proc.-Commun.*, 147, 63–68 (2000).

5. D. G. Jeong and W. S. Jeon, Effective location management strategy based on user mobility classes. *Proc. IEEE, GLOBECOM*, Sydney, Australia, pp. 1426–1430 (1998).

6. Ahmed Abutaleb and Victor O. K. Li. Location update optimization in personal communication systems, *Wireless Networks*, 3, 205–216 (1997).

7. N. Shivakumar and J. Widom, User profile replication for faster location lookup in mobile environments, *First Int. Conf. on Mobile Computing and Networking*, Berkeley, CA, ACM (1995).

8. Joseph S. M. Ho and Jim Xu, History-based location tracking for personal communications networks, *Proc. IEEE Vehicular Technology Conf.* pp. 244–248 (1998).

9. Daqing Gu and Stephen S. Rappaport, A dynamic location tracking strategy for mobile communication systems, *Proc. IEEE Vehicular Technology Conf.* pp. 259–263 (1998).

10. Sajal K. Das and Sanjoy K. Sen, Adaptive location prediction strategies based on a hierarchical network model in a cellular mobile environment, *Comput. J.*, 42, 473–486 (1999).

11. B. Liang and Z. J. Haas, Predictive distance-based mobility management for PCS networks, *IEEE INFOCOM*, pp. 1377–1384 (1999).

12. J. Chan. S. Zhou and A. Seneviratne, A QoS adaptive mobility prediction scheme for wireless networks, *Proc. IEEE GLOBECOM*, pp. 1414–1419 (1998).

13. G. Wan and E. Lin, Cost reduction in location management using semi-real time movement information, *Wireless Networks*, 5, 245–256 (1999).

14. G. Y. Liu and G. Q. Maguire, *A predictive mobility management scheme for supporting wireless mobile computing*, Tech. Report, Royal Institute of Technology, TRITA-IT R 95–04 (1995).

15. G. E. Box and G. M. Jenkins, *Time series analysis*, Holden-Day (1970).

16. G. Peter Zhang, B. Eddy Patuwo and M. Y. Hu, A simulation study of artificial neural networks for nonlinear time-series forecasting, *Computers Op. Res.*, **28**, 381–396 (2001).

17. D. E. Rumelhart and J. L. McClelland (eds), *Parallel distributed processing: Explorations in the microstructure of cognition*, Vol. 1, pp. 318–362, The MIT Press (1986).

18. Simon Haykin, *Neural networks: A comprehensive foundation*, Macmillan (1995).

19. A. Cichoki and R. Unbehauen, *Neural networks for optimization and signal processing*, Wiley (1993).