International Journal of Communication Networks and Information Security

ISSN: 2073-607X, 2076-0930

*Volume **14** Issue **1s** Year **2022** Page **153**:**166***

Role of Neural Network in Mobile Ad Hoc Networks for Mobility Prediction

[1]Vijay U. Rathod, [2]Shyamrao V. Gumaste

[1]Research Scholar, Department of Computer Engineering
MET's Institute of Engineering, Nashik, SPPU, Pune
Maharashtra, India
[2]Professor, Department of Computer Engineering
MET's Institute of Engineering, Nashik, SPPU, Pune
Maharashtra, India
[1]vijay.rathod25bel@gmail.com, [2] svgumaste@gmail.com

Article History

Received: 13 July 2022
Revised: 20 September 2022
Accepted: 30 October 2022

Abstract

The MANETs differ from traditional networks in a lot of aspects, such as high channel error rates, unusual channel features, frequent link breaks, and intense link layer contentions. These characteristics significantly reduce network connectivity, which affects overall network latency, network overhead, network throughput (i.e. the amount of data successfully transferred via a MANETs in a predetermined amount of time), and packet delivery ratio (PDR). For effective network resources preparation and organization in MANETs, the mobility forecast of MN and units is essential. This effectiveness would allow for better planning and higher overall quality - of - service, including reliable facility availability and efficient management of energy. In this research, we suggest to use ELMs, which are renowned for their ability to approximate anything, to model and forecast the mobility of each node in a MANET. Mobility-aware topology control methods and location-assisted routing both leverage mobility prediction in MANETs. It is assumed that each MN taking part in these protocols is aware of its current mobility data, including location, velocity, and movements direction angle. This approach predicts both the locations of future nodes and the distances between subsequent nodes. The interaction or relationship between the Cartesian longitude and latitude of the erratic nodes is better captured by ELMs than by multilayer perceptron's, resulting in mobility prediction that is based on several conventional mobility models that is more precise and realistic.

Keywords: MANETs, Mobility Prediction, ELMs, MLPs

1. Introduction

MANETs are multi-hop wireless networks that may arrange and configure themselves without centralization and in which many mobile nodes link wirelessly and interact naturally in a fast-moving environment. There have been numerous MN implementations in both the civilian and military environments over the previous decades that have focused on MANETs. Two examples of such a deployment are an intelligent oil field or networks of wireless sensors utilized in a network of smart gadgets carried by soldiers on a battlefield. Additionally, MANETs were able to be quickly deployed in unconventional circumstances like disaster recovery because to their capacity for self-organization and self-adaptation without the need for any original infrastructure. In MANETs, every node linked to the network is a peer node with the same features and abilities that enable them to function also as mobile routers. User nodes are often unaware of their future status and have unfettered movement. Although the communication range of nodes in MANETs is limited, they can nevertheless maintain routes and forward packets. As a result, packets are sent through several intermediate nodes in a multi-hop process from source to destination. Because of this distinctive method of message transmission, MANETs depend on node cooperation to function [1].

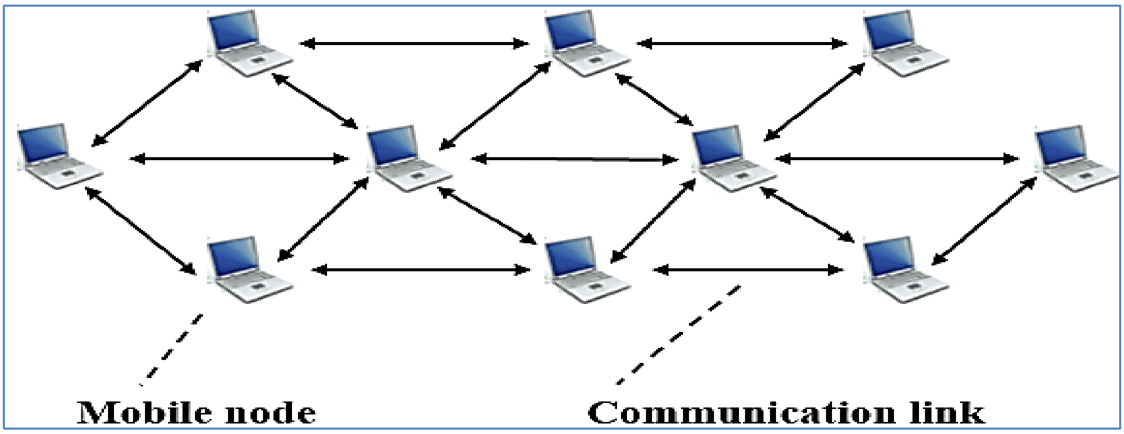

Figure 1. The Generally Recognizable Visual Representations of Manets

1.1. Issues Creation in MANETs due to Node Mobility

Every MN in a MANET has the freedom to travel independently in any direction, therefore if a protocol or previously defined topology changes, the communication link between other nodes may be lost. Dynamic routing requires more energy as a result of higher Packet Loss Ratio (PLR) caused by rising node mobility. Network traffic overload is a common problem for the intermediary node that serves as the network relay. It is preferable to anticipate the next acceptable access point for mobile nodes (MNs) before the user nodes depart its present one in order to create more stable and reliable connections [2].

1.2. Positive Effects of Mobility Management

Mobility prediction is a technique for dealing with issues brought on by node mobility. It does so by anticipating future network topology changes and calculating the trajectory of the MNs' future positions in a dynamic environment to guard against link failure owing to mobility. When compared to fixed wireless systems, the mobility prediction strategy for MANETs is better because it is easier to implement on mobile stations and requires no infrastructure. Our objective in considering these implications is to help MANETs accomplish mobility prediction. In order to effectively plan and manage the bandwidth resources available in wireless networks, it is important to estimate the mobility of wireless users and units. Because of the constant service availability and effective power management that result from this efficiency, planning can be done more effectively, and overall quality of service is increased [3].

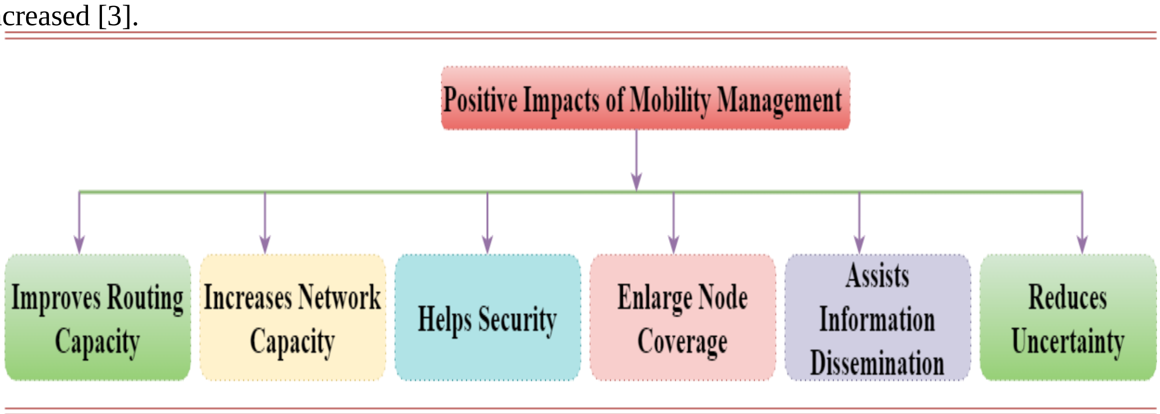

Figure 2.An Overview of the Benefits of Mobility Management

This essay summarizes the advantages of mobility to help readers understand the significance of the positive effects. The old connection-based models are rewritten, and protocols that accommodate for remote mobility are developed. With the use of mobility, we might be able to enhance network capacity, enhance security, and lessen unpredictability.

1.3. Prediction Of Mobility Using Deep Learning

Deep learning is a method that has been put out to predict movements of nodes based on past movements to determine the present mobility of mobile stations depending on pause time, speed, and movement direction. Deep learning uses a deep structure made up of numerous processing layers to create extremely abstract representations of incoming data. Multi-layer neural networks (MNNs), also known as deep learning algorithms, are suggested for this purpose. Since there is still room for improvement, it is necessary to utilize NN to solve problems involving mobility prediction using deep learning.

2. Literature Survey

The challenges with MANETs node mobility are addressed in this research with a neural learning-based approach that effectively predicts future changes in the network structure [4]. The suggested predictor outperforms commonly employed mobility prediction algorithms and achieves accuracy scores that are orders of magnitude higher when applied to both simulated and real mobility traces [4]. The described mobility predictor can enhance the overall QoS in MANETs due to the reached accuracy [4].

The authors of this work recommend using universal approximation ELMs to evaluate and anticipate any node's mobility in a MANET [5]. In MANETs, mobility-aware topology control protocols and location-assisted routing both make use of mobility prediction [5]. Each MN in these protocols is considered to be aware of its current mobility information [5]. In this manner, both future node placements and future node-to-node lengths are predicted [5]. The proposed deep learning technique predicts the current mobility of mobile stations (MSs) based on their halt duration, velocity, and direction of motion. It does this by using the node movement history [6].

From a broad perspective, this chapter examines the effects of MANETs [7]. So avoid being too specific about movement and instead try to paint a bigger picture [7]. This chapter's objective is to support new strategies for utilizing mobility in MANETs based on the present environment and demonstrate how mobility may be beneficial in a variety of ways [7].

Ad hoc networking's service-oriented and application-oriented features may both benefit from mobility prediction [8]. For network-level tasks like call admission control, network resource reserve, service pre-configuration, and QoS provisioning, correct node movement forecast may be necessary [8]. When user movement forecast is integrated with the user's profile, the user may obtain enhanced location-based wireless services at the application level, such as direction suggestions, local traffic flow reports, and online adverts [8]. The most important node motion forecast algorithms for MANETs in the literature are highlighted in this chapter along with their essential design principles and characteristics [8].

The amount of unreliability of the link between the nodes in a MANET rises as a result of node mobility [9]. A link failure can also result in a total route failure, which will impact MANET speed [9]. Therefore, it is necessary to research how node mobility affects the likelihood that a link will break or a route would fail [9]. This study provides a theoretical framework for examining how velocity impacts MANET performance in terms of typical network delay and direction-finding complexity [9].

Researchers in this work thoroughly assess the performance of various mobility handling techniques using single and multiple metric alternatives in an industrial WSN scenario [10]. The results show that in a variety of circumstances, the multiple-metric technique based on fuzzy logic adopted by the researchers outperforms any single metric-based strategy [10].

Using group user trajectory prediction as the foundation, the authors of this study suggested a proactive mobility management strategy [11]. Researchers discuss about a movable user trajectory forecast system that automates the LSTM network and reinforcement learning model training procedure [11]. Researchers are creating a group user trajectory predictor to lessen the computational burden of making predictions for users with similar movement patterns [11].

The authors describe a framework for secure mobility planning for extremely dense edge computing in light of the blockchains reduction in the need for redundant authentication across edge servers [12]. The

mobile handover and service relocation decisions made between base stations are jointly optimized using the Lyapunov optimization, which is then transformed into a multi-objective dynamic optimization process [12].
Here, researchers provide a paradigm for 5G mobile systems mobility prediction. Our research is based on the hypothesis that mobility in vehicular networks has strong correlation, which can be captured by cutting-edge neural network designs to predict the users' point of attachment [13]. Researchers use a combination of Markov chains, recurrent neural networks, and conventional neural networks to demonstrate this, training the networks on mobility trajectories determined by the radio signal received from mobile millimeter-wave devices [13].
The success and spread of the Internet of Mobile Things are still dependent on the management of resources efficiently and mobility [14]. The IoT's overall architecture includes routing as a key component [14]. The current routing protocol RPL is ineffective and susceptible to future research enhancements due to its extremely low sensitivity to mobility. In this study, the RPL protocol has been enhanced to accommodate network mobility [14]. Using the hop count statistic, researchers decreased the amount of hand offs to establish a continuous connection [14]. Due to this, network overhead was decreased and the rate of data delivery was raised [14].
The fundamental network of the fifth generation (5G) is service-oriented [15]. The control plane operations are connected using service-based interfaces, which enhance modularity and are more compatible with cloud networking [15] [16]. In this study, designers present a method for service-oriented radio access networks in which the functionality of next generation application protocol are defined as lightweight services that are simple to test and debug [15]. The handover process is the main topic, and the handover control is presented as a service [15]. S. L. Bangare et al. [17-18] worked in the fields of machine learning and Internet of Things. G. Awate et al. [19] employed CNN techniques. Xu Wu et al. [20] proposed the network security effort. A. S. Ladkat et al. [21] used deep neural networks well for brain tumor research. and colleagues. LMI Leo Joseph et al. [22] have worked real time. The research in [23-25] focuses on newest CNN architecture referred as capsule network. It has dynamic routing process that gives unique output signal to the upper layer regarding output class of the test input. The LSTM variant of this architecture plays vital role in several MANEts.

3. Mobile Node Movement And Prediction Models

3.1. Mobility Models

The usage of the node movement model is essential since it shows how the location, speed, and accelerating of mobile users change over time while also outlining their movement patterns. Utilizing mobility models is essential for accurately replicating the movement patterns of the desired real-world applications. Every mobility model has unique qualities. The initial step in managing mobility is to portray node movement using a realistic mobility model. There are several application situations and varied focuses for various mobility models. The mobility model that has been employed most frequently in recent research papers is summarized in Figure 3 below.

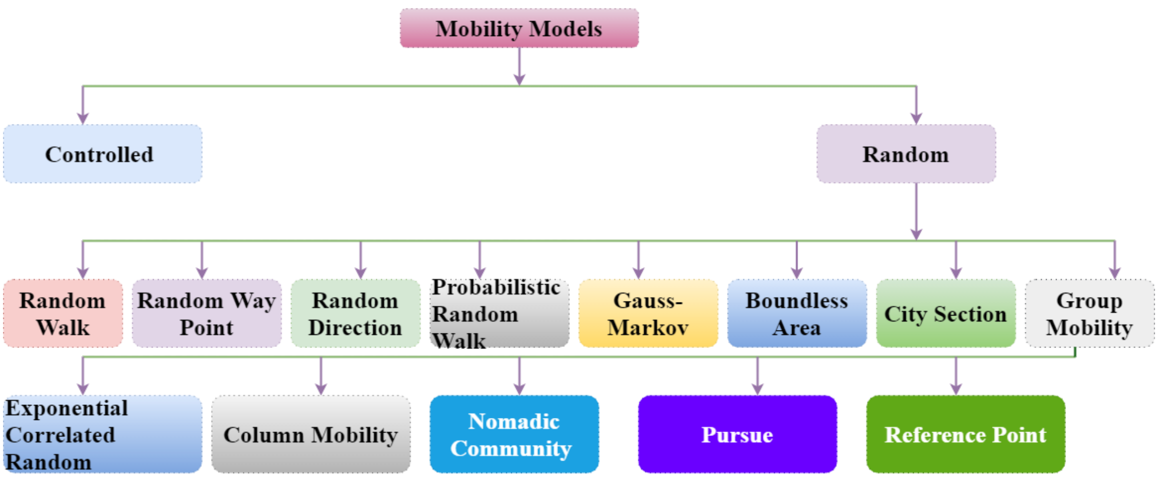

Figure 3. Sorting the Current Mobility Models into Categories

The node movement models in modern research studies can be divided into two categories: I. Controlled Models or Proactive Models. II. Uncontrolled Models or Reactive Models. In controlled models, nodes adjust their trajectories proactively for communication. In addition to models of randomized or uncontrollable mobility, an approach to control the movement of a small proportion of chosen nodes and use this movement to enhance the performance of the network as a whole has been added in recent mobility study works. In the uncontrolled random movement model, researchers have proposed a wide range of ways to represent the natural mobility of nodes. The following are some examples of uncontrolled random mobility models.

3.1.1Mobility Model for a Random Walk

Each node moves by selecting a random direction and speed from a set range to get from where it is to where it needs to be. Such a move is performed for either a constant time or a constant distance traveled. Then a new speed and direction are chosen.

In this mobility model, MN moves from its current location to a new location by randomly choosing both the direction and speed. The new speed and direction are both chosen from ranges defined in advance [*speedmin, speedmax*] and [*0, 2*∏], respectively. The movement can be calculated in two ways: I. either with a constant time interval *t*. II. With a constant distance traveled *d*. The mobility model has a memory-less mobility pattern, meaning that each subsequent move is entirely independent of the one before it.

3.1.2Model for Random Waypoint Mobility

This model incorporates a delay between changes in speed and destination. Firstly, the MN chooses a random location and considers it as its destination and then it moves towards its destination with constant velocity, which is uniformly distributed between [*minvelocity, maxvelocity*]. After arriving at the destination, the MN pauses for a specific time before choosing another random destination. The pause time can have the value zero "*0*", which means that it will continue its movement without any pause. This mobility model also is memory less.

3.1.3Model for Random Directional Mobility

A constant speed and direction must be maintained by the node as it moves toward the simulation area's edge (or until another requirement is satisfied). Then, the node pause and a new direction and velocity are chosen randomly. Then the process repeats.

3.1.4A Model of Area Mobility in Boundless Simulation

With this model, a limitless torus replaces the planar rectangular simulation field.

3.1.5The Gauss-Markov Mobility Model

In this approach, the initial velocity and trajectory of each MN are predetermined. The direction and speed of each MN are updated at predetermined intervals of time *n* to create movement. The velocity and trajectory at the n^{th} instance are determined using the velocity and trajectory at the $(n-1)^{th}$ instance plus a random variable, and are then calculated using the following equations:

$$Sn = aSn - 1 + (1 - a)\,\mu + \sqrt{}\,(1 - a2)\,SXn - 1$$
$$\alpha n = a\alpha n - 1 + (1 - a)\,\mu + \sqrt{}\,(1 - a2)\,\alpha Xn - 1 \quad (1)$$

Where S_n & α_n = MN's new velocity and trajectory at interval *n* respectively, *a*= tuning parameter is used to change the unpredictability, $0<=a<=1$, Random variables S_{Xn-1} & α_{Xn-1} = Gaussian distribution (GD) with a mean of *0* and a standard deviation of *1* respectively. μ has a stable value of *1*. The equation yields values that are entirely random when $a=0$, or Brownian motion. The equation produces fixed values, which are identical to linear motion, for $a=0$. In order to achieve different degrees of random movement, the value of *a* can be varied between these two extremes. Using its present location, speed, and trajectory of movement, the MN calculates its future destination at each time interval. The position of an MN at time intermission *n* is given by the following equations:

$$Xn = Xn - 1 + Sn - 1\,Cos\,\alpha n - 1$$
$$Yn = Yn - 1 + Sn - 1\,Sin\,\alpha n - 1 \quad (2)$$

The X & Y = MN's location at the n^{th} and $(n\text{-}1)^{th}$ time intermission are (X_n, Y_n) and (X_{n-1}, Y_{n-1}) respectively. The MN's velocity and trajectory at the $(n\text{-}1)^{th}$ time interval are S_{n-1} and α_{n-1} respectively.

3.1.6 A Probabilistic Iteration of the Mobility Model for Random Walks

The random walk's previous action in this model affects its subsequent action. When a node moves to the right, there is a greater chance that it will keep moving in that direction than that it will stop. In comparison to the initial random walk model, this causes the walk to depart from the beginning position faster.

3.1.7 Model for City Section Mobility

In this case, a virtual city's street map is merged with the random waypoint movement. These field streets are the only ones that the mobile nodes can travel through. In a comparable concept, voronoi graphs stand in for the streets. Additionally, objects that block radio transmissions are used.

3.1.8 Models of Group Mobility

Group mobility is often added to the models previously discussed, where the behavior of the group is either specified by a function or the nodes are somehow connected to a group target or leader. There is a list of the many group mobility models here.

1. **Model for Exponentially Correlated Random Mobility:** A motion function produces a collective behavior in this instance.
2. **Model for Column Mobility:** The line-shaped set of MNs advances in a specific manner.
3. **Model for Nomadic Community Mobility:** A group mobility model where many mobile nodes move from one location to another simultaneously.
4. **Follow the Mobility Model:** A target node is followed by members of each group as it moves across the simulated area.
5. **Model for Reference Point Group Mobility:** A logical center's path is used as the foundation for the group movement. Once more, the logical centre follows a model of personal mobility.

3.1.9 Mobility Prediction Techniques (MPTs)

In order to effectively deploy agent's node throughout mission time and to fully utilize connectivity, MPTs are used to anticipate the positions of user nodes. The mobility prediction methods are categorized in Figure 4 below depending on the fundamental data used throughout the prediction process. The mobility prediction is performed using a standard position and velocity computation to aid the routing protocol. Ad hoc networks have mobility as one of its intrinsic characteristics. MANETs have no fixed infrastructure and nodes that can move around.

Node migrations often don't affect the application in any way. However, the mobility patterns are frequently necessary for networks to function. Even though each node moves at random, their mobility model nevertheless has some fundamental principles. The first stage in managing mobility is to create and choose a realistic mobility model that accurately represents and forecasts node movement in MANETs.

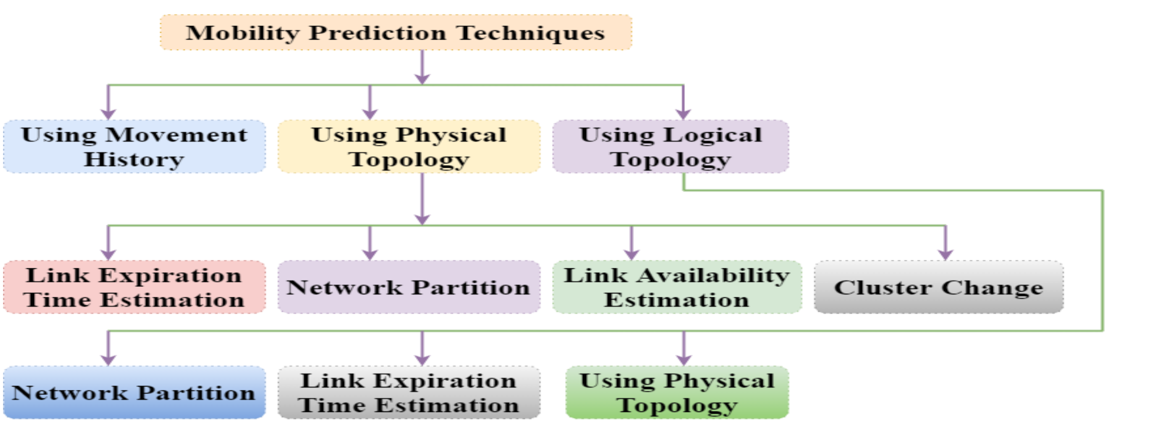

Figure 4. Sorting the mobility prediction Methods into Categories

We divide the mobility prediction techniques for MANETs into the three groups shown in Figure 4:

1. **Using Movement History:** This mobility prediction method predicts a mobile user's future position based on the user's prior movement history or patterns.
2. **Using Physical Topology:** Which make their predictions based on the characteristics of the physical topology of the MANET and which rely on a GPS to establish the specific node position and mobility data?
3. **Using Logical Topology**: This mobility prediction technique selects a logical topology of the MANET to carry out their prediction process (such as a clustering structure). They don't utilize a GPS since, in contrast to the previous category; they don't need precise position and mobility information. Other methods may be employed to acquire estimated values for node location and mobility information (e.g., Inferring the mobility of each node from how the neighborhoods of the node changes over time can be used to estimate internode distances, as can measuring signal attenuation vs. travelled distance).

   For each of the aforementioned categories, we offer well-recognized forecast methods, classifying them into groups based on the applications they are used for or the particular forecast process they are based on.

4. Extreme Learning Machine Improves Node Movement Prediction

A SLFN training algorithm called ELM converges significantly more quickly than traditional methods and yields encouraging results [16]. ELM operates with greater consistency, potency, and accuracy. Regression, clustering, regression, and classification are just a few of the real-time learning issues that ELM has successfully been used to due to its exceptional performance. Mobility prediction systems must be used in wireless networks for effective planning and better QoS. Accurate wireless user and device mobility prediction is made possible by these technologies, allowing for proper coordination and usage of the network communication channel capacity and energy properties. In this paper, the movement patterns of any number of nodes inside a MANET are modeled and predicted using ELMs.

4.1. Theoretical Foundations

This section will demonstrate the theoretical study of fundamental ELM. The most popular artificial neural network structure, SLFNs, was the inspiration for the development of ELM.

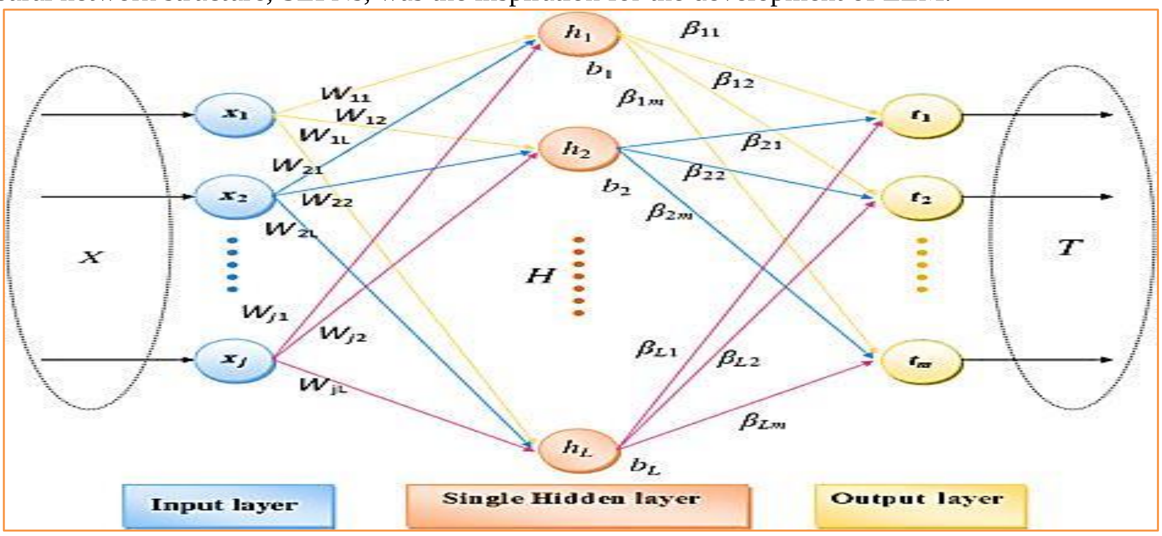

Figure 5. Framework of SLFN

Figure 5 illustrates the three levels of a typical SLFN: I. Input Layer, II. Single Hidden Layer, and III. Output layer. Table 1 contains the notations. Input and output vectors are represented by *x* and *o*. The hidden layer's bias and the weight from the input are represented by *w* and *b*, respectively. β refers to the output weight. The goal of network training is to select the parameters that lead to the best outcome.

4.1.1 SLFN training

The training problem for SLFN will be briefly introduced in this section. Assumed a training set $S=\{(x_i, t_i)| x_i = (x_{i1}, x_{i2},\ldots, x_{in})^T \in R^n, t_i = (t_{i1}, t_{i2},\ldots, t_{im})^T \in R^m\}$, where the input value is represented by x_i and target denotes t_i, the ELM's output *o* with Ñ hidden neurons can be expressed as:

$$Ñ\Sigma i = 1\ \beta ig(WI\ +\ bi)\ =\ Oj,\ j = 1,\ldots\ldots,N \tag{3}$$

Where, *g(x)* stands for the hidden layer's activation function. ELM provides the system with nonlinear mapping by using nonlinear activation functions. Several frequently used activation functions are shown in Table 2.

Table 1. Number of Activation Functions in ELM

Function	Formula
Sigmoid Function	G(a,b,x)= 1/1+exp(-(ax+b))
Hyperbolic Tangent Function	G(a,b,x)=1-exp(-(ax+b))/1+exp(-(ax+b))
Radial Basis Function	G(a,b,x)=exp(-bllx-all)
Multi-quadratic Function	G(a,b,x)=(llx-all+b^2)$^{1/2}$
Hard Limit Function	G(a,b,x)= {1, a.x + b <=0 0,otherwise
Cosine Function	G(a,b,x)= cos (a.x + b)

The purpose of training is to decrease the error between the output of the ELMs and the target. Mean squared error (MSE) is the object function that is most frequently used:

$$MSE\ =\ Ñ\Sigma i = 1(tij - Oij)2, j = 1,\ldots\ldots,m \tag{4}$$

Table 2: Nations

Notation	Meaning
x_i	The input vector of the i^{th} sample
t_i	The target vector of the i^{th} sample
Ñ	The number of hidden nodes in ELM
w_i	The weight vector from input layer to the i^{th} hidden node
b_i	The bias of the i^{th} hidden node
β_i	The weight vector from the i^{th} hidden node to output layer

Where, *N* represents the number of training samples, *i* symbolize the training sample nodes and *j* denote output layer. It can be shown that when *N* approaches infinity, SLFN can approximate all training samples.

$$N\Sigma j = 1\ ll\ Oj\ -\ tj\ ll\ = 0 \tag{5}$$

The universal approximation capacity is satisfied by a set of w_i, b_i, and β_i that is known as:

$$Ñ\Sigma i = 1\ \beta ig\ (wixi\ +\ bi)\ = t,\quad j = 1,\ldots\ldots,m \tag{6}$$

The previous equation can be condensed to

$$\boldsymbol{H\beta}\ =\ \mathbf{T} \tag{7}$$

$$\boldsymbol{H}\ (w1,\ldots\ldots,wÑ,b1,\ldots\ldots,bÑ,x1,\ldots\ldots,xN)\ = \left\{ \begin{array}{c} g(w1x1\ +\ b1)\ldots\ldots\ldots\ldots g(wÑx1\ +\ bÑ) \\ \vdots \\ g(w1xN\ +\ b1)\ldots\ldots\ldots g(wÑxN\ +\ bÑ). \end{array} \right\} \tag{8}$$

$$B= \begin{bmatrix} \beta^T{}_1 \\ \vdots \\ \beta^T{}_Ñ \end{bmatrix}_{Ñxm},\qquad T= \begin{bmatrix} t^T{}_1 \\ \vdots \\ t^T{}_N \end{bmatrix}_{Nxm} \tag{9}$$

Finding the best w_i, b_i and β_i is hence the goal of training the SLFN.

4.2. Fundamentals of ELM

A linear parameter solution and random initialization are the first two steps in an ELM's basic training. Training first stabilizes the random parameters w_i and b_i that ELM uses in its hidden layer.

Table 3. Training of ELM

Training of ELM
Training Set: $S = [\ (x_i, y_i)\ I\ x_i \in R^n,\ Y_i \in R^m,\ i=1,\ldots\ldots,N\ ]$
Initialization: Assign random values to hidden weight w_i and bias b_i and calculate the output the output matrix of hidden layer H using training set.
Analytical Solution: Obtain β from $H\beta=T$ by Moore penrose inverse, $\beta=H^*T$, Where H^* is the Moore-penrose generalized inverse of matrix H

The input vector is transformed into an unpredictably determined feature space with nonlinear activation functions and randomly chosen parameters, which is more efficient than learned parameters. ELM offers the capacity to approximate any function with a nonlinear piecewise continuous activation function. Given that $H\beta=T$ is a linear issue, the Moore-Penrose inverse can be used to find β_i in the second step. Table 3 provides an overview of the ELM training. Without continuously fine-tuning hidden parameters, ELM can produce higher generalization performance.

5. Two Dimensional Long Short Term Memory (LSTM)

Deep learning is now widely employed across many industries, including data processing, speech recognition, semantic comprehension, and picture processing. Deep learning has gained popularity recently in the realm of inertial navigation as well. Deep learning technologies, such as recurrent neural networks and LSTM networks, have a number of advantages over forward networks when simulating nonlinear systems.

The system's capacity for error prediction is enhanced by the usage of 2D LSTM. In this paper, a technique is put forth that makes use of 2D LSTM to estimate positional data based on GPS location data. Through simulation, the value range of the 2D LSTM hyper parameters is investigated with an eye toward the structure of the 2D LSTM.

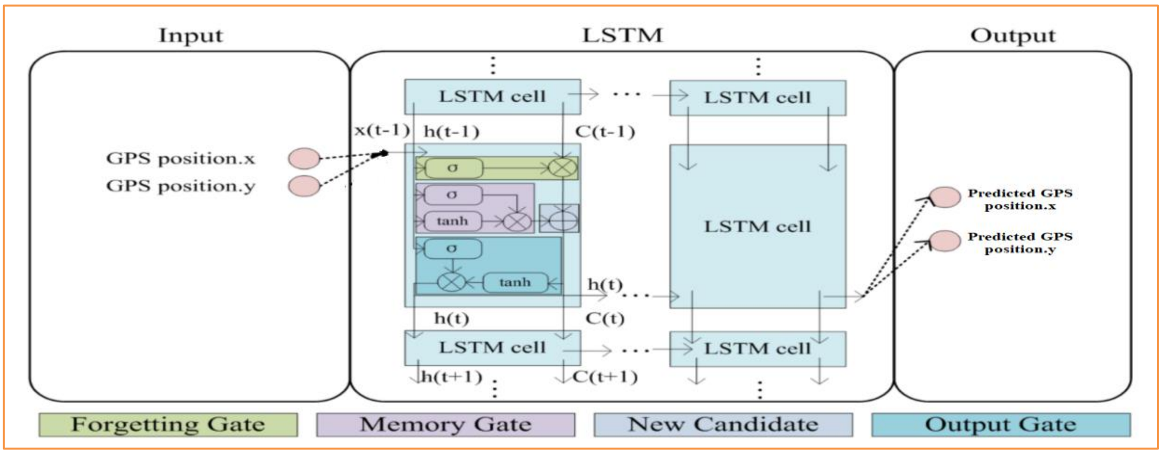

Figure 6. The Future Position Is Predicted Using the 2D LSTM Net Structure

The rounded rectangle in the 2D LSTM cell in Figure 6 represents the neural network layer of the 2D LSTM. The output of an operation is multiplied by a rounded rectangle to produce a gate. Gates are used to regulate the state of the 2D LSTM cell. The repeating module of a 2D LSTM cell is composed of four interconnected layers: a forgetting gate layer, an input gate layer, a new candidate layer, and an output gate layer. A sigmoid function is used in the 2D LSTM net construction to actualize the gate:

$$\sigma(x) = 1/1 + e^{-x} \tag{10}$$

A sigmoid layer creates the forgetting gate layer:

$$f(t) = \sigma\left(W_f[h(t-1), x(t)] + b_f\right) \tag{11}$$

The input gate layer decides what new information will be added to the cell state. There are two steps. A sigmoid layer first selects the values that need to be updated. The state is then expanded upon by a layer of the hyperbolic tangent function *(tanh)*, which generates a vector of additional potential values, $\tilde{C}(t)$.

$$\begin{gathered} i(t) = \sigma(W_i[h(t-1), x(t)] + b_i) \\ \tilde{C}(t) = tan\ h(W_C[h(t-1), x(t)] + b_C). \end{gathered} \tag{12}$$

The new candidate layer is used to update the old cell state, *C (t-1)*, into the new cell state, *C (t)*, as illustrated in the equation below:

$$C(t) = f(t) \times C(t-1) + i_i \times \tilde{C}(t) \tag{13}$$

The location where the output is decided is the output gate layer. First, the decision of the internet to forget earlier will be forgotten. After that, the new candidate values are scaled. We choose the output last. The results are determined by the cell states. The following equations represent the output computation:

$$\begin{gathered} O(t) = \sigma(W_O[h(t-1), x(t)] + b_O) \\ h(t) = O(t) \times tanh\ (C(t)) \end{gathered} \tag{14}$$

The 2D LSTM structure can make use of *i(t)* to decide when to keep or override information in memory cell *C(t)*, and *O(t)* to decide when to access memory cell *C(t)* and when to stop other units from being disturbed by *C(t)*. The information is output as *O(t)*, and the LSTM structure information is output as *h(t)*.

The forgetting gate in the structure allows the 2D LSTM net to achieve conditional predictions that depend on time. In other words, the 2D LSTM learns the properties of location information connected to time while learning a huge amount of GPS position data. The GPS determines the details of the current inertial position based on the previous step. The GPS integrated navigation system, which uses *KF* to estimate position information, is currently built on the final step estimation. The processed ones can be compared to a time-based conditional prediction. In light of this, we may estimate location data using LSTM and GPS position data.

6. Results And Performance Analysis

The training rate is 0.8, as seen in Figure 7. This shows that 80% of the input dataset is made up of the training set, and the remaining 20% is made up of the test set. The 2D LSTM is used in practice to determine input and output size.

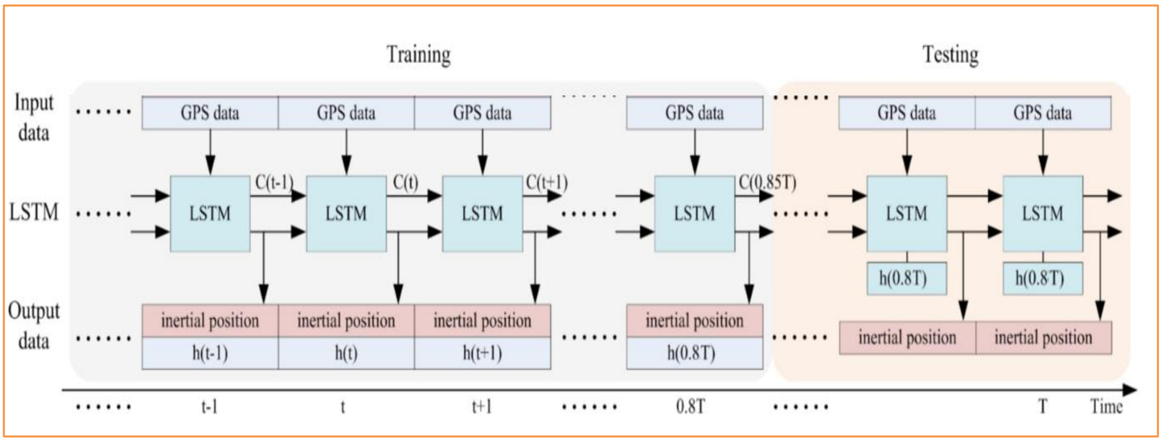

Figure 7.The 2D LSTM Is Trained And Tested To Estimate Positional Information.

Figure 8. Gauss-Markov Mobility Prediction Using an ELM Model (Training Phases)

The input set includes GPS data, which includes positional data in two dimensions, as seen in Figures 6 and 7. The training target data are the GPS data and the 2D LSTM input. The output dataset contains both the two-dimensionally projected inertial location and the output of the two-dimensionally LSTM, with the number of dimensions based on the two-dimensional LSTM structure.

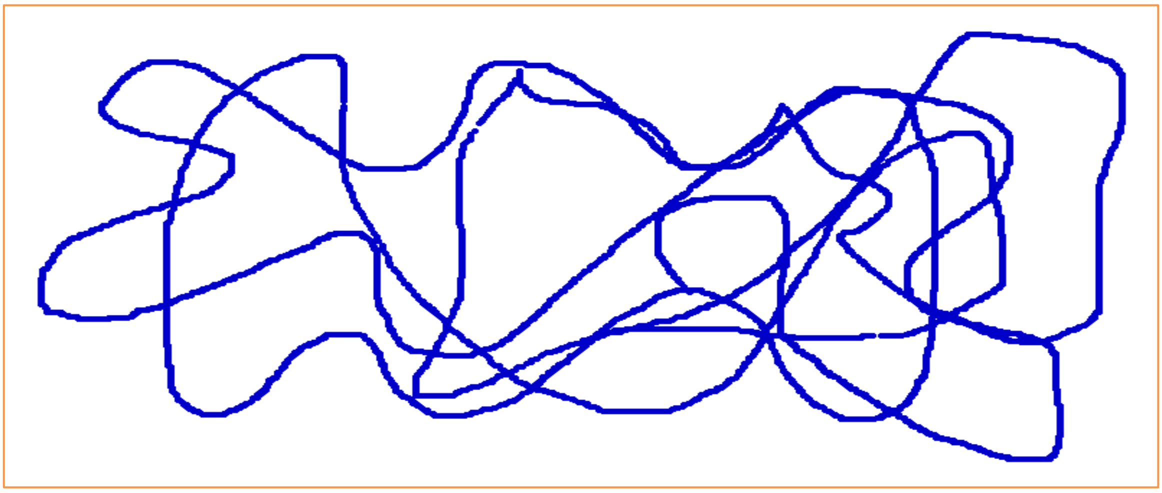

Figure 9. Gauss-Markov Mobility Prediction Using an ELM Model (Testing Phases)

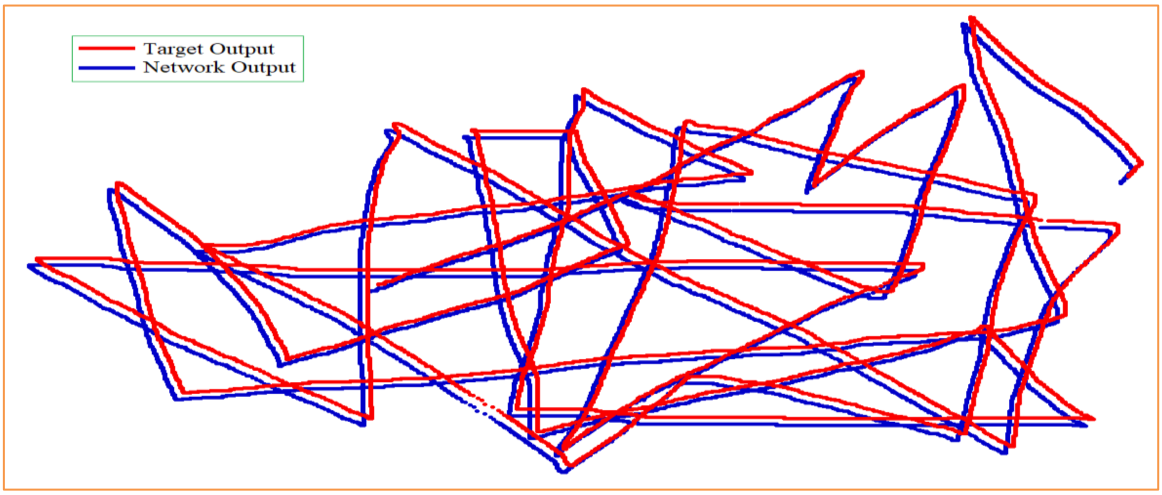

Figure 10. Using MLP, Predict Two Mixed Mobility Modes (Training Phases)

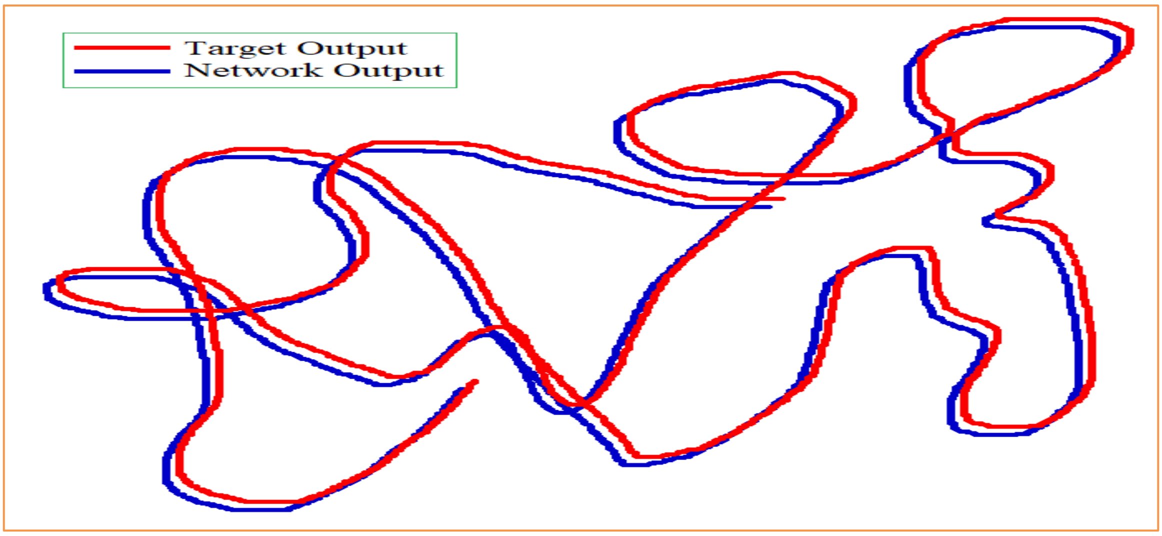

Figure 11. Two Mixed Mobility Modes Are Predicted Using MLP (Testing Phases)

Table 4. The Statistical Analysis between 2D LSTM, Random Forest and Geometric Progression

Datasets	Parameter Matrix	Proposed 2D-LSTM	Random Forest	Geometric Progression
Dataset 1	Mean Absolute Error (MAE)	5.5	16.57	13.715
	Root Mean Squared Error (RMSE)	5.5227	13.94	13.23
Dataset 2	Mean Absolute Error (MAE)	2.5	7.53	6.234
	Root Mean Squared Error (RMSE)	2.5495	6.44	6.11
Dataset 3	Mean Absolute Error (MAE)	3.5	10.54	8.728
	Root Mean Squared Error (RMSE)	3.5355	8.93	8.47
Dataset 4	Mean Absolute Error (MAE)	4.5	13.56	11.222
	Root Mean Squared Error (RMSE)	4.7434	11.98	11.36
Dataset 5	Mean Absolute Error (MAE)	4.5	13.56	11.222
	Root Mean Squared Error (RMSE)	5.1478	13	12.33
Dataset 6	Mean Absolute Error (MAE)	1.5	4.52	3.741
	Root Mean Squared Error (RMSE)	1.5811	3.99	3.79
Dataset 7	Mean Absolute Error (MAE)	3.5	10.54	8.728
	Root Mean Squared Error (RMSE)	4.3012	10.86	10.3
Dataset 8	Mean Absolute Error (MAE)	1.5	4.52	3.741
	Root Mean Squared Error (RMSE)	1.5811	3.99	3.79

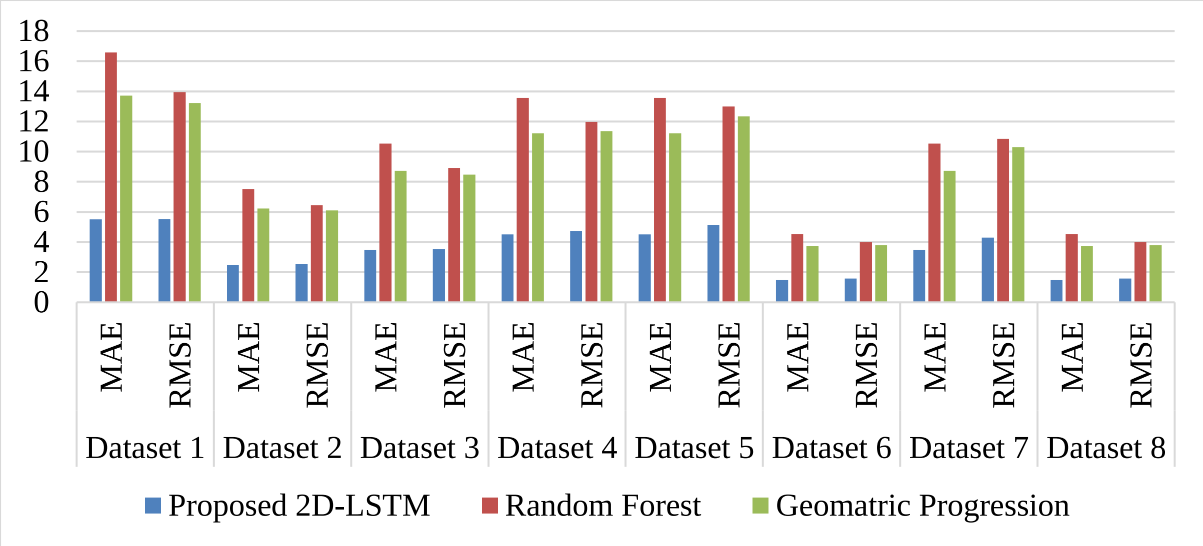

Figure 12. The Performance Analysis (2D LSTM VS Random Forest VS Geometric Progression)

7. Conclusion

We suggest an innovative approach for predicting node mobility in a MANET in this paper. The cornerstone of the suggested solution is the ELM, which is architecture with a single feedforward layer. In contrast to MLPs, ELMs don't need any parameter tuning, and the performance of the predictions is unaffected by the initial weights. Additionally, ELMs better capture the interaction and correlation between the arbitrary nodes' Cartesian coordinates, resulting in more precise and realistic mobility predictions based on a variety of conventional mobility models. The simulation results are utilized to demonstrate how the recommended prediction method can significantly outperform conventional methods based on MLPs. In a further study, the recommended prediction method will be developed to predict routing tables, which will reduce the volume of data transferred in MANETs and lengthen the battery life of the node.

Reference

[1] V. U. Rathod and S. V Gumaste, "Effect Of Deep Channel To Improve Performance On Mobile Ad - Hoc," *J. Optoelectron. Laser*, vol. 41, no. 7, pp. 754–767, 2022.

[2] N. P. Sable, V. U. Rathod, P. N. Mahalle, and D. R. Birari, "A Multiple Stage Deep Learning Model for NID in MANETs," *2022 Int. Conf. Emerg. Smart Comput. Informatics, ESCI 2022*, pp. 1–6, 2022, doi: 10.1109/ESCI53509.2022.9758191.

[3] N. P. Sable, V. U. Rathod, P. N. Mahalle, and P. N. Railkar, "An Efficient and Reliable Data Transmission Service using Network Coding Algorithms in Peer-to-Peer Network," no. August, pp. 144–154, 2022.

[4] L. Ghouti, "Mobility prediction in mobile ad hoc networks using neural learning machines," *Simul. Model. Pract. Theory*, vol. 66, pp. 104–121, 2016, doi: 10.1016/j.simpat.2016.03.001.

[5] L. Ghouti, T. R. Sheltami, and K. S. Alutaibi, "Mobility prediction in mobile Ad Hoc networks using extreme learning machines," *Procedia Comput. Sci.*, vol. 19, no. Ant, pp. 305–312, 2013, doi: 10.1016/j.procs.2013.06.043.

[6] Y. Yayeh, H. P. Lin, G. Berie, A. B. Adege, L. Yen, and S. S. Jeng, "Mobility prediction in mobile ad-hoc network using deep learning," *Proc. 4th IEEE Int. Conf. Appl. Syst. Innov. 2018, ICASI 2018*, pp. 1203–1206, 2018, doi: 10.1109/ICASI.2018.8394504.

[7] F. Li, Y. Yang, and J. Wu, "Mobility Management in MANETs: Exploit the Positive Impacts of Mobility," pp. 211–235, 2009, doi: 10.1007/978-1-84800-328-6_9.

[8] D. Gavalas, C. Konstantopoulos, and G. Pantziou, "Mobility Prediction in Mobile Ad-Hoc Networks," *Next Gener. Mob. Networks Ubiquitous Comput.*, no. January, 2011, doi: 10.4018/9781605662503.ch021.

[9] M. Das, B. Sahu, and U. Bhanja, "Mobility and its effect on the performance of MANET,"

2015 IEEE Power, Commun. Inf. Technol. Conf. PCITC 2015 - Proc., no. i, pp. 871–877, 2016, doi: 10.1109/PCITC.2015.7438118.

[10] Z. Zinonos and V. Vassiliou, “Mobility management solutions in industrial wireless sensor networks,” *Proc. - 15th Annu. Int. Conf. Distrib. Comput. Sens. Syst. DCOSS 2019*, pp. 675–681, 2019, doi: 10.1109/DCOSS.2019.00121.

[11] Z. Zhao *et al.*, “Reinforcement Learning Trajectory Prediction,” vol. 17, no. 4, pp. 2102–2116, 2020.

[12] H. Zhang, R. Wang, W. Sun, and H. Zhao, “Mobility management for blockchain-based ultra-dense edge computing: A deep reinforcement learning approach,” *IEEE Trans. Wirel. Commun.*, vol. 20, no. 11, pp. 7346–7359, 2021, doi: 10.1109/TWC.2021.3082986.

[13] F. Meneghello, D. Cecchinato, and M. Rossi, “Mobility prediction via sequential learning for 5G mobile networks,” *Int. Conf. Wirel. Mob. Comput. Netw. Commun.*, vol. 2020-Octob, pp. 0–5, 2020, doi: 10.1109/WiMob50308.2020.9253407.

[14] S. Hoghooghi and R. N. Esfahani, “Mobility-Aware Parent Selection for Routing Protocol in Wireless Sensor Networks using RPL,” *2019 5th Int. Conf. Web Res. ICWR 2019*, pp. 79–84, 2019, doi: 10.1109/ICWR.2019.8765273.

[15] E. Pencheva, I. Atanasov, I. Asenov, and V. Trifonov, “Mobility management as a radio access network service,” *2019 10th IFIP Int. Conf. New Technol. Mobil. Secur. NTMS 2019 - Proc. Work.*, pp. 1–5, 2019, doi: 10.1109/NTMS.2019.8763816.

[16] S. Sumathi and S. Paneerselvam, “Computational Intelligence,” *Comput. Intell. Paradig.*, pp. 25–52, 2020, doi: 10.1201/9781439809037-6.

[17] S. L. Bangare, A. R. Khare, P. S. Bangare, “Code parser for object Oriented software Modularization”, International Journal of Engineering Science and Technology, ISSN: 0975-5462, Vol. 2 (12), 2010, 7262-7265.

[18] S. L. Bangare, “Classification of optimal brain tissue using dynamic region growing and fuzzy min-max neural network in brain magnetic resonance images”, Neuroscience Informatics, Volume 2, Issue 3, September 2022, 100019, ISSN 2772-5286, https://doi.org/10.1016/j.neuri.2021.100019.

[19] Gururaj Awate, S. L. Bangare, G. Pradeepini and S. T. Patil, “Detection of Alzheimers Disease from MRI using Convolutional Neural Network with Tensorflow”, arXiv, https://doi.org/10.48550/arXiv.1806.10170

[20] Xu Wu, Dezhi Wei, Bharati P. Vasgi, Ahmed Kareem Oleiwi, Sunil L. Bangare, and Evans Asenso, “Research on Network Security Situational Awareness Based on Crawler Algorithm”, Security and Communication Networks, Hindawi, ISSN:1939-0114, E-ISSN:1939-0122, vol. 2022, Article ID 3639174, 9 pages, 2022. https://doi.org/10.1155/2022/3639174.

[21] Ajay S. Ladkat, Sunil L. Bangare, Vishal Jagota, Sumaya Sanober, Shehab Mohamed Beram, Kantilal Rane, Bhupesh Kumar Singh, "Deep Neural Network-Based Novel Mathematical Model for 3D Brain Tumor Segmentation", Computational Intelligence and Neuroscience, vol. 2022, Article ID 4271711, 8 pages, 2022. https://doi.org/10.1155/2022/4271711.

[22] LMI Leo Joseph, S. L. Bangare et al., "Methods to Identify Facial Detection in Deep Learning Through the Use of Real-Time Training Datasets Management", EFFLATOUNIA - Multidisciplinary Journal, vol. 5, no. 2, pp. 1298-1311, ISSN 1110-8703.

[23] Sandeep Pande and Manna Sheela Rani Chetty, “Analysis of Capsule Network (Capsnet) Architectures and Applications”, Journal of Advanced Research in Dynamical and Control Systems, Vol. 10, No. 10, pp. 2765-2771, 2018.

[24] Pande S.D., Chetty M.S.R. (2021) Fast Medicinal Leaf Retrieval Using CapsNet. In: Bhattacharyya S., Nayak J., Prakash K.B., Naik B., Abraham A. (eds) International Conference on Intelligent and Smart Computing in Data Analytics. Advances in Intelligent Systems and Computing, vol 1312.

[25] Sandeep Pande and Manna Sheela Rani Chetty, “Bezier Curve Based Medicinal Leaf Classification using Capsule Network”, International Journal of Advanced Trends in Computer Science and Engineering, Vol. 8, No. 6, pp. 2735-2742, 2019.