

Received March 7, 2021, accepted March 17, 2021, date of publication March 22, 2021, date of current version March 30, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3067816

Prediction of Scenarios for Routing in MANETs Based on Expanding Ring Search and Random Early Detection Parameters Using Machine Learning Techniques

DURR-E-NAYAB¹, MOHAMMAD HASEEB ZAFAR^{1,2}, (Senior Member, IEEE),
AND ALI ALTALBE²

¹Faculty of Electrical and Computer Engineering, University of Engineering and Technology, Peshawar 25120, Pakistan

²Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia

Corresponding author: Durr-e-Nayab (nayab.khan@uetpeshawar.edu.pk)

ABSTRACT Routing protocols in Mobile Ad Hoc Networks (MANETs) play a pivotal role in ensuring quality of service (QoS) and improving network performance. Selection of optimal routing protocol and suitable parameters for a given network scenario is a major task that ultimately affects the behavior of network. This work exploits machine learning (ML) techniques for the selection of adequate routing parameters and protocol by regression of parameters in given network scenario to ensure optimal performance. The network is trained based on parametric setup of expanding ring search mechanism (ERS) and random early detection (RED) technique to estimate network throughput, end to end (E2E) delay and packets delivery ratio (PDR) and is tested via wide-ranging simulations in varying network topologies. Both RED and ERS mechanisms are aimed to control link and node level congestion in the reactive routing protocols and our aim is to select the best suited parameters for given network topologies based on ERS and RED parametric setups and improve performance for ensuring QoS. ML algorithms are trained and tested for their performance in varying network topologies. We have exploited these models with best performance for ERS and RED based routing in given topological arrangements. The performance of the ML algorithms is evaluated on the basis of root mean squared error (RMSE) and mean absolute error (MAE) for regression settings. Prediction models with up to par RMSE and MAE out-turns are attained and exploited for selection of suitable ERS and RED parameters and routing protocols in order to ensure the QoS for given network scenario. Variants of standard routing protocols are devised based on their performance and the ML techniques are exploited for prediction of QoS parameters to decide on the optimal variant that attains significant improvement in performance. Results are shown to confirm that considerable improvement in QoS is attained.

INDEX TERMS Ad hoc multi hop wireless networks, congestion control, expanding ring search, machine learning, mobile ad hoc networks, on demand routing protocols, random early detection, regression, quality of service.

I. INTRODUCTION

In recent research, ad hoc routing has taken over the networking domain due to its high end applications in internet of things (IoT), wireless ad hoc sensor networks (WASN), vehicular edge computing (VEC), and ad hoc networks (VANETs) and unmanned mobile vehicles (UMV) [1]–[3]. Researchers are more interested in playing

The associate editor coordinating the review of this manuscript and approving it for publication was Eyuphan Bulut².

with the ad hoc environment to optimize the network based on routing parameters. Mobile ad hoc networks (MANETs) has also gained great attention due to its highly dynamic nature where the network comprises of nodes that carry out the communication without any fixed infrastructure through vibrant ways and performing as source nodes, destination nodes or even as intermediate nodes depending upon the requirement of the network [4]. The connectivity is managed through hops by hops communication without any base station due to restricted transmission range of the wireless edges. Hence

managing the routing mechanisms in ad hoc networks has become a critical job and plays a pivotal role in affecting the Quality of Service (QoS) [5].

We addressed the problem of congestion due to flooding in reactive routing protocols of MANETs. The reactive routing protocols maintain the routes on on-demand basis for which they exchange frequent route update messages. Since reactive routing protocols do not maintain big routing tables like proactive routing protocols, they constantly keep exchanging route updates. Hence the motivation for this work is to tackle the issue of congestion due to flooding in reactive routing with optimizing the packets handling at the network level and retransmissions at the transport level by exploiting the expanding ring search mechanism and routing parameters. This is attained by changing the way in which the packets are handled for routing. Mechanisms are designed for this purpose in which the TTL value of each route request is associated with per hop behavior before processing the packet. ML algorithms are exploited for this purpose and the QoS metrics are improved as shown in the results section.

When nodes are in the transmission range of each other, the link between them is up and they can communicate directly. Contrarily if the nodes are not in the transmission range of each other i.e. the link between them is down then the source node can take facility of intermediate nodes in order to reach its destination [4]. The number of intermediate nodes specifies the number of hops required for the message of the source node to reach the destination node that also stipulates the path distance between the source and destination in terms of hops. This way of communication where nodes are exploited in hop by hop fashion for routing is also known as mobile multi hop ad hoc networks [5]. Fig. 1 depicts a typical MANET environment where the source node A is communicating with destination node B via various intermediate nodes in their transmission range.

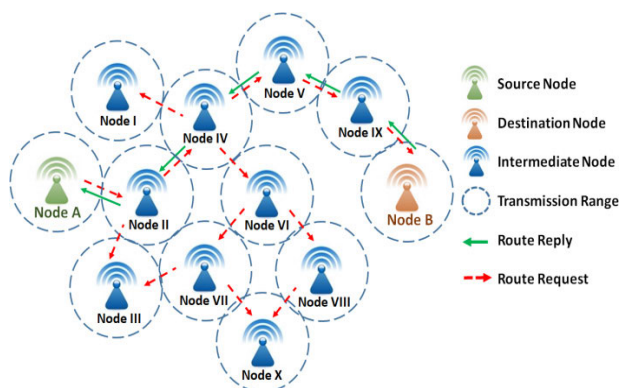


FIGURE 1. Source node communicates with target node by means of intermediate nodes in a MANET environment.

Nodes in MANETs do not perform periodic tasks neither are they allotted fixed jobs to keep records for long period of time to cover the whole network, hence routing in MANETs becomes really complicated task due to this dynamicity. Due this hop to hop communication and non fixed responsibility

of nodes numerous messages and requests are repeatedly sent from source node that may lead to congestion and ultimately affecting the network performance due to flooding. Nodes with predefined time to live (TTL) for message request are initialized and once the nodes starts searching for destination the TTL value is decremented for each hop until the TTL is exhausted [6]. There are also numerous other parameters associated with the routing mechanism of MANETs such as number of route requests, expiry time, traversal time, route retries, waiting time, hop count, network diameter, TTL thresholds and many other parameters that decides how the routing behaves and responds in different network situations [7]–[9]. Automating these parameters according to network scenario is a crucial but very affective task which can bring drastic improvement in routing and ultimately improve network performance by timely tackling issues like congestion. Performing this job manually can be really tricky and no effective work is available that deals with such scenario.

A. EXPANDING RING SEARCH TECHNIQUE

Many routing protocols in MANETs make use of methods like expanding ring search (ERS), random early detection (RED) and many other techniques and their variants to exploit these routing parameters in various formats in order to ensure QoS and deal with issues like congestion due to flooding, energy drainage, overheads and link breakage [10]–[20]. In ERS the route search is managed in the form of rings of predefined steps instead of linear search methods. ERS has a mechanism for searching the destination node in terms of hop counts that expands in predefined steps while monitoring the TTL values. ERS uses the TTL field in the IP header of the node to conduct the route search in non conventional manner. The radius of search expands stepwise until the target is reached or the whole network is traversed. Other parameters such as TTL start, TTL threshold, TTL increment, route retries, time to expire, buffer time, network traversal time, network diameter, node traversal time etc are also monitored accordingly in order attain efficient routing [11]. The details of these parameters are given in the Request for Comments (RFCs) by the internet society such as Internet Engineering Task Force (IETF) for each routing protocol that exhibit ERS [21], [22].

Fig. 2 provides a general illustration of ERS mechanism where node A (source node) is trying to reach node B (destination node). The search steps are taken in the form of rings as in ERS-Ring-1, ERS-Ring-2, ..., ERS-Ring-5. The nodes have to be in the transmission range of each other in order to communicate and progress from source node A to destination node B through intermediate nodes. ERS mechanism resolves the issue of flooding but network may lead to counter overheads and loads that cause energy drainage in highly dense networks [23]. To tackle such issues many version of ERS enhancements [10]–[13] has been proposed that manages the route search with modified parametric setups and techniques to improve routing and QoS.

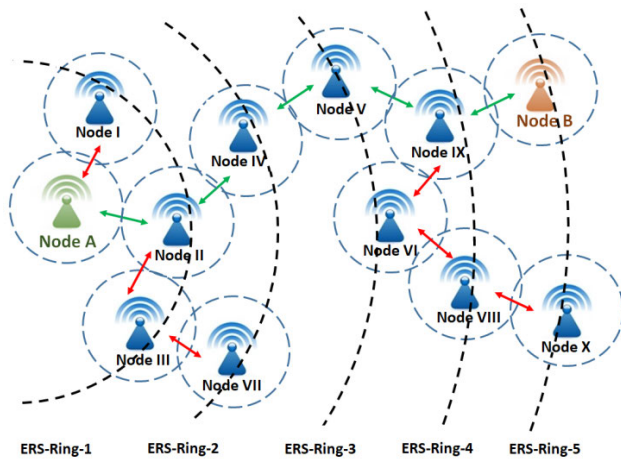


FIGURE 2. An illustration of Expanding ring search mechanism in MANET routing.

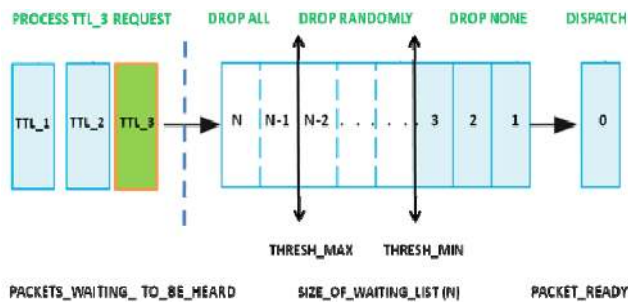


FIGURE 3. An illustration of random early detection mechanism in MANET routing.

B. RANDOM EARLY DETECTION TECHNIQUE

Techniques akin to RED also manage to reduce the load on the network in case of congestion because of flooding or high route queries making use of parametric setups as discussed earlier. RED and its variants such as Weighted Random Early Detection (WRED), Adaptive Random Early Detection (ARED), Robust Random Early Detection (RRED), Smart Random Early Detection (SRED) and many others [14]–[20] execute packets dropping that are wasting more network energy and taking longer to be processed for routing in a pre-emptive manner. Instead of conventional manner where the packets are dropped irrespective of any criteria or priority when the buffer is full, RED manages the packets drop before the buffer is full on pre-emptive premises.

A characteristic illustration of RED mechanism is provided in Fig. 3 where two thresholds (i.e. THRESH_MAX and THRESH_MIN) are exploited for deciding the mechanism of drop the packets. Three packets (i.e. TTL_1, TTL_2 and TTL_3) are coming towards the buffer with size N and are queued directly unless THRESH_MIN is reached. When the number of packets in buffer exceeds THRESH_MIN, the packets are dropped randomly at defined rate. All the packets are dropped once the THRESH_MAX is reached. Packet 0 is already ready to be processed for routing. There are several parameters such as IP precedence, DSCP, buffer size etc are exploited by RED based on which the packets are dropped before congestion occurs [24], [25]. We take the concepts of

these techniques and exploit the ML techniques to automate the parametric setup of routing protocols of MANETs.

In this work we have exploited the same parametric setups and the way in which the packets are presented to the routing protocols such that the ML techniques are trained on these networks data to automate the parametric setups of MANETs with respect to the network scenarios. The models are used for decision making in case of congestion and other critical network condition to make the right choice for parametric setups and select the suitable routing mechanism. The major contributions of this work are training the ML algorithms on ERS and routing parameters of MANET under random way point (RWP) mobility and using regression to automate the optimal parametric setup of these quantities in order to attain improved QoS. The parameters are not only automated but the ERS is used to design a mechanism for packets handling in such a way that the packets are handled with different priorities, drop rates, congestion flags and threshold depending upon the requirement of the network. This idea of handling packets on ERS parameters is exploited with the RED technique that is never done before and proposed by us in [34].

The rest of paper is organized into **Section II** that provides details of the related work, **Section III** provides an insight on the ML techniques exploited for experimentations, **Section IV** provides details on metrics used for ML algorithms implementation, **Section V** demonstrates the methodology and stepwise implementation of the proposed algorithm, **Section VI** illustrates the simulation environment, **Section VII** discusses the results and observations and **Section VIII** concludes the paper and provides an insight on future enhancements.

II. RELATED WORKS

There are several techniques that are proposed for improving QoS by congestion control for routing in MANETs based on parametric setup and ML techniques. Guo *et al.* [5] proposed a delay prediction mechanism integrated with a proactive ad hoc network routing protocol namely OLSR. They used queuing delay only and showed that queuing delay can be modeled as a non-stationary time series. They used multi-layer perceptron (MLP) and radial basis function (RBF) to predict from the non-stationary time series model of queuing delay in MANET. Ghadimi *et al.* [6] proposed an analytical model to predict accurate media access delay by obtaining its distribution function in a single wireless node. They claim to derive accurate analytical models for the media access delay for IEEE 802.11 ad hoc networks in finite load conditions with and without exposed terminals. Variations of ERS techniques i.e. Blocking and Improved Blocking ERS are proposed in [10] and [11] that exploit the parameters such as different waiting time to manage early flood cancellation and attain high performance. Another technique proposed in [12] exploits the route request (RREQ) parameter and the TTL_INCREMENT values to reduce the energy consumption by the nodes and improving the QoS in terms of PDR,

E2E delay and throughput. A packet priority mechanism based on network coding scheme is proposed in [13] that takes in to account a class based priority to the ERS rings and uses proportional fair scheduling to deal with congestion. The ML technique MLP was exploited for WRED method to automatically adapt the end users to the network and improving the QoS [14]. Per hop behavior (PHB) parameter was exploited to ensure the QoS for new users in the network. The technique improved the performance of the network and enabled the network to respond efficiently in manual QoS parameter pre-settings. Authors in [18] presented a probability based RED technique to tackle ill-behaved traffic flows by managing the buffer inflow and outflows. The authors claimed significant performance improvement by dropping the packets on well defined probability model. An unequal packet priority parameter is exploited for Smart RED in [20] for its application in TCP to UDP traffic with different needs of bandwidth inspired by smart access point with limited advertised window (SAP-LAW) concepts. The research in [26] proposed a hybrid technique for clustering and queuing is attained by arranging the packets in queues before been processed for routing. Differentiated services parameter was used for setting the priority of packets and selecting them for drop in case of congestion. The priority is allotted on the basis of buffer aided decodable network coding.

Guo and Malakooti [27] presented a scheme for predicting mean delays using neural network a time series using tapped-delay-line MLP network and tapped-delay- line Radial Basis Function Network (RBFN). The inputs used by them were mean delay time series itself only and the mean delay time series together with the corresponding traffic loads. They ignored the effect of any other parameter on delay as well as their scheme predict only one hop delay not the complete end-to-end packet delay. The model proposed in [28] devised a Q-learning algorithm for improving MANETs routing using reinforcement methods. The technique is exploited for automating the routing decisions based on reinforcement concepts. Continuous Hopfield neural network (CHNN) model was proposed for Dynamic Source Routing (DSR) in [29] to obtain the most stable route to improve the QoS in MANET. Authors in [30] proposed different methods for mobility prediction. However, these methods assume that nodes move according the Random Waypoint Mobility (RWM) model. As a result, nodes mobility prediction moving according to other models can lose its accuracy and efficiency. Mobility prediction allows estimating the stability of paths in a mobile wireless ad hoc network. Hongyan *et al.* [31] used autoregressive models and neural network to predict internet time delay whereas Tabib and Jalali [32] used feed-forward multilayer perceptron for the same purpose.. Both considered only internet time delay and have not considered any other network types and their characteristics. An adaptive QoS routing is proposed in [33] based on prediction for link performance in MANETs. The predictions are made for lower layer parameters in order to attain QoS and improve mobility.

Routing in MANETs is an interesting research domain due to extremely dynamic nature of MANETs and its applications in copious domains. The motivation for this work is to explore ML mechanism for existing routing protocols and those proposed in [33] and [34] to enhance the performance of MANETs on QoS premises. This is attained by changing the way in which various parameters are defined for routing in case of congestion and normal scenarios.

III. MACHINE LEARNING TECHNIQUES EXPLOITED FOR REGRESSION SCENARIOS OF QoS

A. LINEAR REGRESSION

Linear Regression (LR) is a very simple type of supervised ML algorithm that solves only regression problems. LR estimates the coefficients for the hyper plane that separates the input data on optimal premises. LRs are estimated based on model parameters that are predictable from the input data. In simple LRs two variables are used for finding the predictive function i.e. the predictor variable which the independent variable and the criterion variable which is the dependant variable [35]. We have exploited LR algorithm for forecasting the routing parameters as the predictor variables to generate a predictive model based on observed routing dataset. The model is exploited for ensuring the QoS based on trained routing data set and additional values of the variables are used to test the trained model for predicting an unknown response. For our data we have removed the highly correlated input attributes in order to achieve higher efficiency. Similarly irrelevant attributes to the output variable are also removed with feature selection methods. In order to reduce the complexity of the model ridge regularization technique is used that averts any coefficient to reach high value by reducing the absolute sum square of the learned coefficients [35].

LRs have very simple implementation and representation as they combine the set of single inputs such as terrain size, nodes speed or network density to achieve particular outputs such as E2E delay, throughput or PDR both of which are related based on linear relationship or linear model. The LR model is also exploited for multiple inputs such as buffer size, time to expire, thresholds and upper bounds for predicting the outputs as mentioned for single inputs scenario. Once the model is developed, making predictions is as simple as solving an equation for a particular set of inputs. A simple LR model for single variable scenario can be represented as in (1).

$$Y(t) = B_0 + B_1^*X(t) \quad (1)$$

where $X(t)$ is the input dataset, $Y(t)$ is the output dataset, B_0 is the bias coefficient and B_1 is the coefficient for $X(t)$. The aim is to find the optimal value for the coefficients that relate the input with the output. In case of multiple input variables refer to (2) where $X_1(t) \dots X_n(t)$ are the input variables datasets and $B_0 \dots B_n$ and $B_1 \dots B_m$ are the bias coefficients and coefficients for relating in inputs.

$$Y(t) = B_0 + B_1^*X_1(t) + B_2^*X_2(t) + B_3^*X_3(t) + \dots + B_n^*X_n(t) \quad (2)$$

```

yy < 1900
| yy < 1300
| | yy < 500 : 99.91 (2/0.01)
| | yy >= 500
| | | yy < 700 : 98.76 (1/0)
| | | yy >= 700 : 98.06 (3/0)
| yy >= 1300
| | yy < 1500 : 94.6 (1/0)
| | yy >= 1500 : 92.48 (2/0.01)
yy >= 1900
| yy < 2300
| | yy < 2100 : 90.82 (1/0)
| | yy >= 2100 : 88.97 (1/0)
| yy >= 2300
| | yy < 2700 : 85.97 (2/0.01)
| | yy >= 2700
| | | yy < 2900 : 84 (1/0)
| | | yy >= 2900 : 82 (1/0)

Size of the tree : 19
    
```

FIGURE 4. An illustration of Decision Tree model for MANET problem.

B. K-NEAREST NEIGHBORS

k-Nearest Neighbors (KNN) is also a supervised ML algorithm that can solve both regression and classification problems. KNN uses whole dataset for training in classification problems whereas in regression problems it utilizes the features similarity method for predicting the new data point [36]. KNN chooses k nearest data points where k can be any integer depending upon the data i.e. for bigger data the value of k is larger and vice versa. In case of regression problem, KNN takes the mean of k most related instances in the training data. We have used the automated value for k using cross validation. Euclidean distance is used to measure the distance between the data because the routing data is in same scale where as Linear NN search method is exploited for the way in which the data is searched and stored [37].

C. DECISION TREE

Decision tree exhibits a tree like model of decisions that comprises of branches of consequences of event outcomes and resource costs and utility associated with them. Decision tree also solves both regression and classification problems that is why they are referred to as classification and regression trees (CART) [38]. The tree is constructed on the basis of greed based selection of suitable split points for predictions that is repeated until the depth is reached. We have devised an adaptive method for specifying depth the decision tree depending upon the problem. An internal node of a branch is a test on an attribute and the branch is the outcome of given node or test. The leaf node represents the decision finalized after computation of the attributes which is also known as class label. The classification rules are defined through the paths from root to leaf of the tree [39]. We have used the model to prune to generalize to new routing data. A simple illustration of decision tree is given in Fig. 4 where the decision for PDR estimation has to be taken based on the terrain size length variable (YY) with a tree of size nineteen. The output illustration in the form of tree is given in Fig. 5.

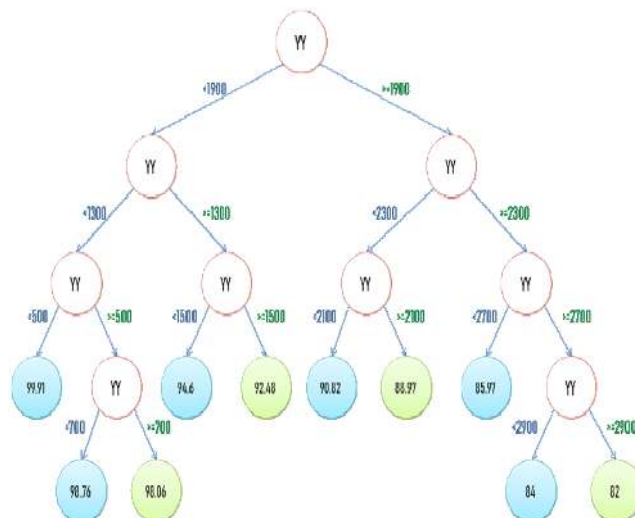


FIGURE 5. Output representation of decision tree for MANET problem.

The figure shows that how the decision tree develops solution for a simple regression scenario by generating branches of possible output possibilities.

D. SUPPORT VECTOR MACHINE

Support vector machines (SVMs) also known as support vector networks (SVNs) are another type of supervised learning algorithm that support classification problems in particular and has an adaptation to support regression problems known as support vector regression (SVR). The algorithm is based on statistical framework that solves linear problems typically and non linear problems using kernel trick. The SVR solves the regression scenario by attaining a line of best fit that reduces the error of cost function via optimization process that considers only the support vectors i.e. the data instances closest to the line [40]. The best fit line or hyper plane is not always a straight line as per problem scenario; hence in some cases a line with curves and polygonal regions is achieved using different kernels. Fig. 6 depicts an SVM model for typical regression problem. The model shows input data that in our case is the routing data fed to the model that applies the support vectors (SVs) for extracting the required output values in given scenario. The alphas (α_n) represents the weights given to the output of each SV and a bias is applied on the final output that is used for toning the output accordingly.

We evaluated standard reactive routing protocol AODV (V-ERS) with the varying parametric setups used versions of AODV i.e. AODV1g, AODV2g, AODV3g, AODV4g and AODV5g, AODV1s, AODV2s, AODV3s, AODV4s and AODV5s in which 1,2,3,4,5 are configuration versions and g and s are the gigantic and small modes based on the topological format. The gigantic terminology is also associated with huge networks with higher requirement for QoS that implies the extension of these algorithms to 5g, 4g and 3g networks. Among these formats the best ones V3g, V4g and V5g are selected as they attain tangible enhancement. ML algorithms are exploited for all routing protocols and

TABLE 1. ERS and RED parametric setups for V5g, V4g, and V3g in varying network topologies.

Version	Mode	Time_Exp	Route_Ret	Wait_Time	Size_Buff	Thrsh_1	Thrsh_2	Thrsh_3	UP_B1	UP_B2	UP_B3	Drop_Rate
V5g	MOD1	30	2	10	36	10	10	15	30	35	30	15-25
	MOD2	30	5	12	15	8	5	10	16	30	25	8-25
V4g	MOD1	30	2	22	30	10	10	15	15	35	30	5-25
	MOD2	24	6	16	32	8	5	10	15	30	25	7-25
V3g	MOD1	24	2	12	32	14	10	15	25	35	30	10-25
	MOD2	20	6	14	15	8	5	10	15	30	25	7-25

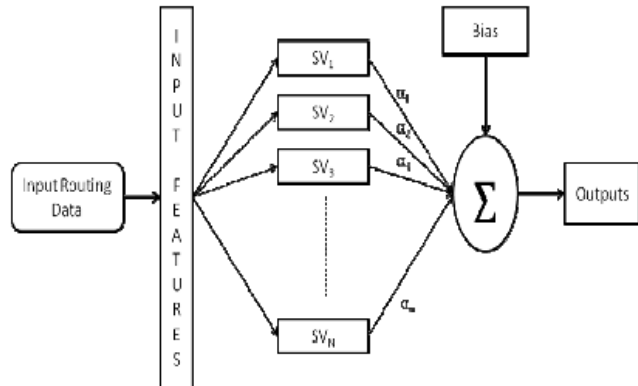


FIGURE 6. Typical support vector regressor (SVR) model representation.

their variants based on Table 1 configurations. It is important to mention that the selections are made through ML and the fixed optimal configurations and application of random early detection on expanding ring search parameters are also proposed by our research and is never done before. Also as shown in Table 6, 7, 8 and 9 the ML algorithms are exploited for the standard routing protocols as well as the modified versions of reactive routing protocol with the configurations of Table 1. V-ML is the reactive routing version selected by the ML algorithms with best performance. Fig. 14 shows the performance of these algorithms and improvement with the proposed algorithms. It is visible that the ML is selecting the optimal routing paradigm (V-ML) on the basis of QoS metrics.

We have tested the performance for linear, polynomial and RBF kernel to fit the observed data and selected the best suited kernel for given test data. The learning rate is set to be 0.001 and the regularization parameter is set to be 1/epochs hence the higher the number of epochs (i.e. 10^{12}) the lower is the regularization parameter. We have exploited polynomial kernel method instead of linear kernel method in order to attain better performance. Polynomial kernel prediction method is expressed mathematically in (3) where input (x) and support vectors (y) are used to calculate the estimations setting the value of p that is the degree of polynomial and c that is a free parameter to trade off the influence of polynomial between higher and lower order.

$$K(x,y) = (x,y)^p \text{ or } K(x,y) = ((x,y) + c)^p \quad (3)$$

E. ARTIFICIAL NEURAL NETWORK

Artificial neural networks (ANNs) are complex models with huge configuration parameters tuned for given classification

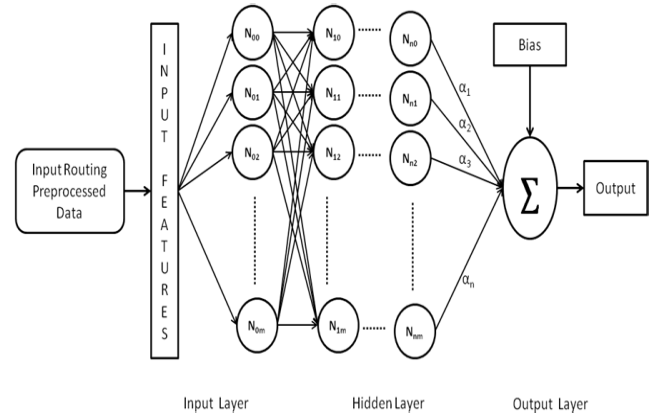


FIGURE 7. Typical artificial neural network (ANN) model representation.

or regression problem. The parameters are configured through exhaustive process of learning and numerous trial and error checks. The ANN model we have exploited for our data is a feed forward ANN (FFANN) that is also known as a multi layer perceptron (MLP) that is majorly composed of three layers i.e. input layer, hidden layer and output layer. This model takes an analogy from the natural neural networks where the building blocks i.e. neurons are arranged in layers to perform approximations [41], [42]. Then number of hidden layers, number of neurons and activation functions are selected for given problem based in performance analysis and trial and error method. This process can also be managed through evolution as done in evolutionary ANN but we have explored simple feed forward neural network. The model utilizes back propagation supervised method for training that learns through updating the connection weights after each run. The learning rate is set to be 0.3, momentum 0.2, validation threshold 20 and the number of epochs is set to be 500 initially.

Fig. 7 depicts a typical FFANN model representation. The model consists of input layer that is the instances of the input features ($Att_1, Att_2, \dots, Att_n$), the hidden layer transform the values from the input layer with weighted linear summation ($\alpha_1, \alpha_2, \dots, \alpha_n$) followed by an activation function that takes the output from the last hidden layer and transform into output values and toning with a bias. During the back propagation process for training the model the parameters are updated using the gradient of the loss function with respect to parameters that needs adaptation. MLP uses the square error loss function expressed mathematically in (4).

$$Loss(x,y,W) = 0.5||x - y||_2^2 + 0.5*\alpha||W||_2^2 \quad (4)$$

where a $\|W\|_2^2$ is the L2 regularization term that is used as penalty to control the error and α is the non-negative hyper parameter [43]. MLP is very sensitive to feature scaling and it performs well when the data is scaled well.

IV. METRICS FOR MACHINE LEARNING TECHNIQUES

A. INPUT FEATURES FOR MACHINE LEARNING TECHNIQUES

1) NETWORK DENSITY

The network size in terms of total number of nodes in network defines the network density. The network density has effect on the E2E, throughput and PDR as denser networks may cause congestion, signal interferences and retransmission and vice versa. We have used network density as input to our ML models and observe the response of the network in terms of E2E delay, throughput and PDR. The network density is varied from 5 to 100 with varying node to sink ratio.

2) MAX NODE SPEED

Max nodes speed specifies the upper velocity limit up to which the nodes can move. The nodes velocity also has significant effect on the QoS. Very high or very low speeds tend to negatively effect on E2E delay, throughput and PDR and vice versa. We have taken the speeds ranging from 1 to 100 mps for our experiments taking into account the normal speed human, ground and air vehicles.

3) TERRAIN SIZE

The physical area of the network, taken in x and y coordinates and denoted in meter squares is the terrain size in which the nodes are scattered. The terrain size also has impact on the QoS as the larger terrains tend to scatter the nodes far away from each other whereas too small terrains tend to congest the nodes and impact E2E delay, throughput and PDR likewise. We have observed the terrains ranging from $300 \times 500\text{m}^2$ to $3000 \times 3200\text{m}^2$.

B. OUTPUTS FOR REGRESSION SCENARIO OF MACHINE LEARNING TECHNIQUES

1) END TO END DELAY

The time taken by a packet to reach from source node to destination node over the network intermediate nodes is the E2E delay. The E2E delay is the sum of all delays such as request processing delays, buffering delays of route discovery, retransmission delays, queuing delays, propagation and transfer delays etc. Average E2E delay for n packets can be measured using (5) that is the average sum of difference between the time a packet is received and the time when the packet is sent. E2E delay measures the capability of a specific protocol and architecture used to communicate between nodes besides noise profile and media access mode. In simulations the delay is measured in nano seconds that is then converted into

seconds.

$$E2E_{avg} = \sum_{i=0}^n \left(\frac{\text{ith Packet Time Received} - \text{ith Packet SentTime}}{\text{Total Number of Packets Received } (n)} \right) \quad (5)$$

2) THROUGHPUT

The average data rate or rate of packets being received at destination nodes from source nodes is the throughput of the network. Throughput is also sometimes measured as the bandwidth of the channel. We have measured throughput in kbps using (6). The bytes are converted to bit to be tallied in kbps and if packets are used then measured in number of packets received.

Throughput

$$= (\text{Bytes_Received} * 8) / (\text{LastPacket.GetSeconds}() - \text{FirstPacket.GetSeconds}() * 1024) \quad (6)$$

3) PACKETS DELIVERY RATIO

The measure of successful delivery of packets from source to destination is the Packets Delivery Ratio (PDR). This metric is also used to measure the efficiency of the network protocol or architecture exploited. If the PDR is low that may create the scenario of congestion due to retransmissions or incomplete data transmissions. The PDR is measured using (7) that illustrates PDR mathematically.

$$PDR = \text{Packets_Received} / \text{Packets_Transferred} * 100 \quad (7)$$

C. PARAMETERS AUTOMATION FOR MANET ROUTING

The ultimate goal of this work is to automate the mentioned parameters and more parameters for improved performance of MANETs routing. We have done both the automation of parameters and improvement of QoS metrics with ML algorithms. Among these parameters, those for ERS such as thresholds, drop rates and upper bounds are already automated and selected. But since the ranges for these parameters were manually selected from the predictions of simulations results in NS-3 and then training the ML algorithms on same values imply partial automation, hence we have claimed and used automated values. The regression of these parameters is done with ML algorithms for improving QoS parameters, which is the aim of this paper and that is why we have focused on these results. The automation of all mentioned and more parameters is an exhaustive task in process and requires respective details, methods and explanations that we aim to contribute as future enhancement of this research as mentioned in the conclusion section. The details of parameters that we are automating are as follows:

1) SIZE OF BUFFER

Every node in the network has a queuing buffer that stores the packets before being processed for routing. When nodes in the network send packets at higher rate and the processing

of packets takes longer time then the buffer overflow occurs due to congestion that may cause loss of packets before being queued. The maximum limit for a node buffer is already specified by the routing protocol i.e. max buffer limit for AODV is 64 and that for DSR is 50 packets. The size of buffer can be calculated using mathematical expression given in (8).

$$Size_Buff_{Remaining} = Length_Buff_{Max} - Packets_Buff_{Occupied} \quad (8)$$

where $Size_Buff_{Remaining}$ is the buffer capacity, $Length_Buff_{Max}$ is the max buffer limit as defined by routing protocol and $Packets_Buff_{Occupied}$ is the number of packets already occupying the buffer.

2) BUFFER OUTSTANDING TIME

When a packet reaches an intermediate or target node it is first queued in to the node buffer. The packet has to wait for buffer outstanding time ($T_{Outstanding}$) before being handled by routing protocol for routing. The packet is allowed to wait till max buffer time after which the packet is dropped for retransmission. The buffer outstanding time also known as waiting time or queue buffering time is responsible for causing queuing delays. When packets have longer buffer outstanding time due to congestion or any other reason, it may lead to further worsening of the situation in the form of increased congestion and packets drop or denial of services (DoS). Buffer outstanding time is mathematical illustrated in (9).

$$T_{Outstanding} = T_{Max} - T_{Arrival} - T_{Current} \quad (9)$$

where $T_{Outstanding}$ is the residual time of a packet, T_{Max} is the total allowed waiting time (i.e. 30s for DSR, DSDV and AODV), $T_{Arrival}$ is the arrival time of the packet and $T_{Current}$ is the time at which the residual time is being calculated.

3) NODE ROUTE ATTEMPTS

The sender node attempts to send a packet on route towards target once or a number of times depending upon the fact that the target is reached in single or multiple attempts. When a node does not receive a route reply within specified time it retransmits the packet unless it receives a reply and the packet reaches its target. Allowing infinite retransmissions may lead to congestion in case the target is unreachable and the network density is high. Hence the routing protocols in MANET specify fixed number of retransmissions that are allowed for a packet in order to reach its destination. Each time the node attempts to retransmit the copy of original packet the route attempt is incremented as illustrated in (10).

$$Route_Ret(N_j) = Route_Ret(N_j) + 1 \quad (10)$$

4) NUMBER OF HOPS

The number of hops is the node by node steps taken by a packet when it is dispatched by sender towards the destination. A count is kept by the packet header that is incremented each time a packet travels from one node to another. Higher hops count illustrates that the packet as spent more time in

the network and more packets with higher hops count results in high bandwidth consumption that ultimately results in congestion at transport layer. The number of hops is evaluated as illustrated mathematically in (11) of how when a packet moves from one node (N_i) to another (N_j).

$$Hop_Count(N_j) = Hop_Count(N_i) + 1 \quad (11)$$

5) PACKET EXPIRY TIME

Each packet when starts traversing the network for its destination, remains in the network for specified time known as packet expiry time after which the packets leaves the network. The packets expiry time is specified by the routing protocol at the time when the packet leaves the source node and keeps on decrementing until the packets either reaches the target node or leaves the network. The packet expiry time can be expressed mathematically as shown in (12).

$$T_{Expiry} = T_{Start} + T_{TTL} - T_{Current} \quad (12)$$

where T_{Expiry} is the packet expiry time, T_{Start} is the time when packets leaves the source node, T_{TTL} is the total time of the packet in the network and $T_{Current}$ is the current time of the packet in the network.

V. PROPOSED METHODOLOGY

In order to implement our algorithms for attaining better performance for routing in MANETs, we exploit open source simulator Network Simulator-3 (NS-3) [44]–[46] to implement the topologies that generates the routing data as inputs for ML algorithms training and testing phases. Initially MANET's scenarios are implemented to monitor the performance of existing routing protocols such as AODV, DSR, DSDV and OLSR in terms of E2E delay, throughput and PDR. Once enough data is generated after exhaustive simulations, modified ERS and RED mechanisms are implemented with the parametric setup as shown in Table 1 for various topologies and monitored again for E2E delay, PDR and throughput. Routing data of three of the best scenarios i.e. V5g, V4g and V3g are used for our ML techniques training and testing. These variants are the reactive routing protocols versions with the Table 1 configurations. Two modes of each version are produced based on the fact that the networks behave differently in different topological arrangements such as network densities, max node speeds and terrain sizes.

Different parametric setups are prepared for adverse topologies (MOD1) where the routing performance of the network decline whereas in normal topologies (MOD2) the network already performs well [47]–[49]. Other parameters include time to expire ($Time_Exp$), route retries ($Route_Ret$), buffer waiting time ($Wait_Time$), size of buffer ($Size_Buff$), thresholds for congestion detection ($Thrsh_1$, $Thrsh_2$ and $Thrsh_3$), upper bounds for packets dropping (UP_B1 , UP_B2 and UP_B3) and packets drop rates ($Drop_Rate$) [34]. The details and mathematical explanations for all these parameters are given in Section VI (A). Fig. 8 provides an illustration of thresholds, drop rates and upper bounds

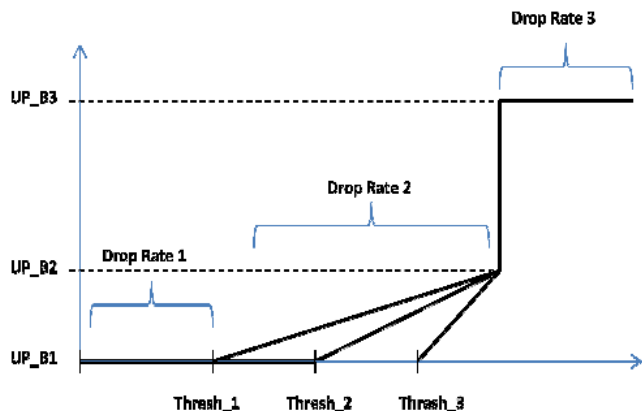


FIGURE 8. Illustration of parametric setup of thresholds, upper bounds and drop rates for RED algorithm.

employed for RED implementation. The values are taken from ERS implementation of AODV. Fig. 9 provides an illustration of the proposed methodology in the form of block diagram. Initially MANET topology is implanted in NS-3 and routing data is gathered from existing standard routing protocols and their modified versions based on ERS and RED parameters. Important features are extracted from the data for training the ML algorithms and the ML techniques are tested for their performance.

We have done both the automation of parameters and improvement of QoS metrics with ML algorithms. Initially we have selected the best suited fixed parameters manually under given network setups or modes. Among these twelve parameters, those for random early detection such as thresholds, drop rates and upper bounds are already automated and selected through heuristic algorithm. Since the ranges for these parameters were manually selected from the predictions of simulations results in NS-3 and then training the ML algorithms on same values imply partial automation. It is important to mention that the fixed configurations in Table 1 for routing are attained manually through simulations and are proposed by our research [34]. These fixed configurations perform well with respect to the network modes or setups in which they are employed and vice versa. These configurations proposed by us perform well in their respective network modes but to automate their selection we have employed ML algorithms. This is how we first automate these fixed configurations to improve the QoS metrics and later we are working to automate all parameters irrespective of any fixed configurations as an extension of this work. Hence we have claimed and used automated values and in order to relate with title and major contribution of the manuscript i.e. prediction of scenarios, the ML algorithms are trained on network parameters and the fixed configurations. The regression of these parameters is done with ML algorithms for improving QoS parameter that is the aim of this paper and that is why we have focused on these results.

The performance of each ML algorithm is evaluated for each protocol separately for varying network topologies and the best performing techniques are then exploited for future

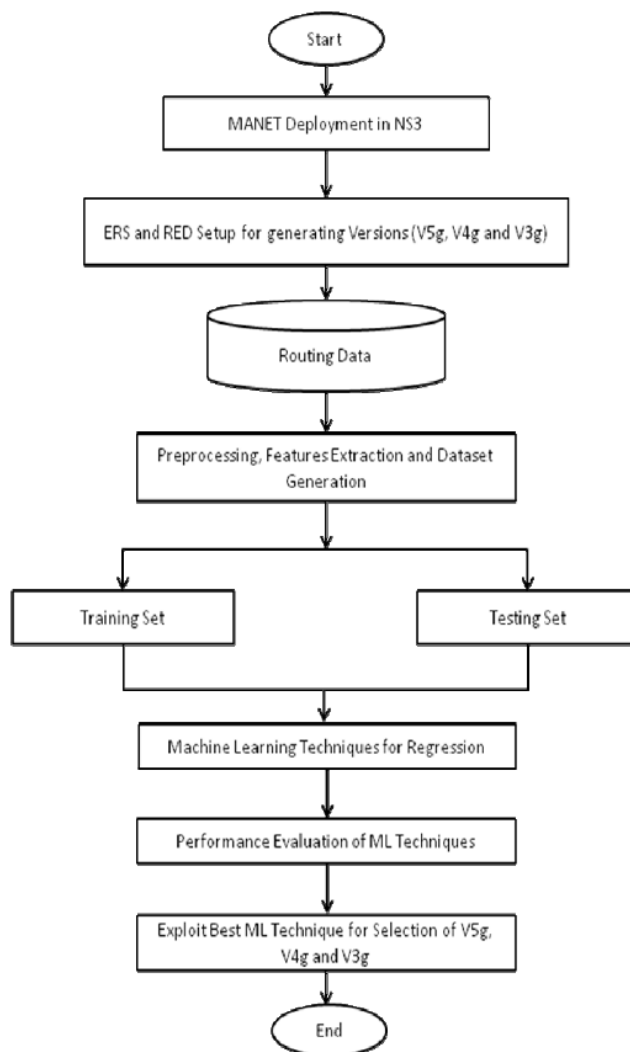


FIGURE 9. Block diagram of the proposed methodology.

estimations. These estimations are used for regression scenarios to predict the outputs for each protocol in terms of E2E delay, throughput and PDR in order to select the best suited routing protocol among monitored developments. The estimations are then used for testing to predict the performance of the protocols in situations under consideration and protocol with optimal performance is selected. The ML results are monitored with actual simulation results for evaluations.

VI. EXPERIMENTAL SETUP AND SIMULATION ENVIRONMENT

A. SETUP FOR MACHINE LEARNING ALGORITHMS

For all ML methods 10 folds cross validation is used for input data to create the model. Almost 675 instances are used for throughput and 294 instances are used for E2E delay and PDR. For LR model M5 method is used for most attribute selection while in some cases greedy model is used depending up the performance and a batch size 100 is used. For our data we have removed the highly correlated input attributes in order to achieve higher efficiency. Similarly irrelevant attributes to the output variable are also removed with feature

TABLE 2. Specifications of E2E delay data used for experiments.

Param	Protocol	Dmin	Dmax	Davg	Dsdv
Speed	OLSR	20.5	37	24.4221	5.04046
	AODV	2.39	5.7	3.39563	0.78826
	DSDV	32	38.4	36.27	1.6268
	DSR	10.5	27.78	16.1484	6.59124
	AERSAODV	1.2	3.7	1.70358	0.51251
Terrain	AODV	1.84	23.6	8.0014	6.616157
	DSDV	2.39	5.53	4.034	0.917814
	DSR	3.36	32.75	14.2653	7.231316
	OLSR	1.54	16.81	7.4854	4.8121
	AERSAODV	1.22	2.78	2.12467	0.52853
Node Density	AODV	13.7	29	23.4489	5.1581
	DSDV	2.39	7.3	4.33111	1.59117
	DSR	15.7	112	60.2944	33.3052
	OLSR	22.1	27.3	24.7756	1.84283
	AERSAODV	2.3	13.2	8.32111	4.42428

selection methods. We have used the automated value for k using cross validation. Euclidean distance is used to measure the distance between the data because the routing data is in same scale.

Linear NN search method is exploited for the way in which the data is searched and stored. Linear search mechanism is utilized for searching the nearest neighbor and no windowing is exploited for any problem. We have tested the performance of SVM for linear, polynomial and RBF kernel to fit the observed data and selected the best suited kernel for given test data. The learning rate is set to be 0.001 and the regularization parameter is set to be 1/epochs hence the higher the number of epochs (i.e. 10^{12}) the lower is the regularization parameter. We have exploited polynomial kernel method instead of linear kernel method in order to attain better performance. The learning rate of ANN is set to be 0.3, momentum 0.2, validation threshold 20 and the number of epochs is set to be 500 initially. Table 2, 3 and 4 tabulate the specifications of E2E delay, throughput and PDR data respectively used for running the ML experiments. The data specifications are given in terms of minimum values (Dmin, Thmin, Pmin), maximum values (Dmax, Thmax, Pmax), average (Davg, Thavg, Pavg) and standard deviation (Dsdv, Thsdv, Psdv). The data has been normalized for experiments using (13), where $X = \{x_1, x_2, x_3, \dots, x_n\}$ is the set of input data and Norm(x_i) is the i^{th} normalized data item.

$$\text{Norm}(x_i) = \frac{x_i - \min(X)}{\max(X) - \min(X)} \quad (13)$$

B. SETUP FOR NETWORK SIMULATIONS

We have implemented our MANET topologies using an open source discrete event network simulation tool NS-3 that is extensively exploited wireless Ad-hoc networking research. The details of parametric setup and considerations for MANET topologies are provided in Table 5. We have used varying topologies to generate routing data for training our ML techniques; hence some of the parameters are taken as input features and are varied for different scenarios. Parameters that are kept constant are tabulated in Table 5.

TABLE 3. Specifications of PDR data used for experiments.

Param	Protocol	Thmin	Thmax	Thavg	Thsdv
Speed	AODV	15.09	22.87	19.3274	2.44418
	DSDV	17.46	19.93	18.499	0.65576
	DSR	12.54	18.74	16.3205	1.82713
	OLSR	17.55	23.1	20.9921	1.51595
	AERSAODV	20	23.4	22.13	0.92598
Terrain	AODV	1.53	20.48	12.20133	8.34176
	DSDV	1.51	20.48	11.6694	8.14652
	DSR	1.6	20.49	11.45887	7.94432
	OLSR	1.65	20.48	11.85833	8.24163
	AERSAODV	3	20.5	13.34533	7.55755
Node Density	AODV	5.88	48.03	25.4367	15.3516
	DSDV	5.69	41.17	22.73	13.4963
	DSR	5.9	35.58	20.7033	11.4091
	OLSR	5.83	46.7	24.7708	14.5724
	AERSAODV	5.45	53	27.1292	17.3871

TABLE 4. Specifications of PDR data used for experiments.

Param	Protocol	Pmin	Pmax	Pavg	Psdv
Speed	AODV	68.34	95.94	83.6542	8.722636
	DSDV	85.71	94.76	92.7958	1.969736
	DSR	50.98	76.22	64.7505	7.603388
	OLSR	58.34	85.94	73.7068	8.713711
	AERSAODV	86.3	98.94	96.8084	1.322024
Terrain	AODV	79.2	100	90.3641	7.683799
	DSDV	82.8	96.95	92.1107	4.838711
	DSR	7.64	100	52.5709	37.46521
	OLSR	82	100	92.6717	6.146138
	AERSAODV	82	99	95.5273	2.593562
Node Density	AODV	80.6	85.34	82.9233	1.469728
	DSDV	86	90.07	88.7156	1.2235
	DSR	69.49	70.83	70.14	0.442804
	OLSR	83.4	88.5	85.3744	1.614629
	AERSAODV	87	96	91.5556	3.32081

C. METRICS FOR PERFORMANCE EVALUATION

1) MEAN ABSOLUTE ERROR

The performance of the ML techniques is evaluated on the basis of Mean Absolute Error (MAE) that is the measure of error in paired observations for specified problem scenario [51]. In a given input set of k instances, X_i' are the estimated values for given observations X_i , the MAE can be mathematically expressed as shown in (14).

$$\text{MAE} = \left(\frac{1}{k}\right) \sum_{i=1}^k |x_i' - x_i| = \left(\frac{1}{k}\right) \sum_{i=1}^k |e_i'| \quad (14)$$

MAE is the arithmetic sum of absolute errors ($e_i' = |x_i' - x_i|$) that measures the average magnitude of errors where x_i' is the prediction and x_i is the true value. MAE is a linear scores i.e. all the differences are measured equally while taking the average.

2) ROOT MEAN SQUARED ERROR

Root Mean Squared Error (RMSE) is also used for measuring the performance of ML techniques in terms of standard deviation of the estimated errors [51]. RMSE is the measure of the distance of the data from the line of best fit. RMSE can

TABLE 5. Parametric setup for implementing topologies in NS-3.

Parameter	Value	Description
Transmission Model	ns3::FriisPropagationLossModel	Friis lossless model presumes that the propagation of nodes occur in free space without obstruction.
Transmission Power	7.5dB	Tx is initialized for each node that reduces over time.
Wifi Mode/Application Rate	2 MB	Network bandwidth with constant bit rate traffic.
Network Settling Time	50s to 100s	Time required for network to reach normal state.
Communication pause time	0s	Time break taken for communication between the nodes.
Simulation Time	1200s	Each simulation is run for 1200s or approx 20 minutes.
Runs	20	Each simulation is tested for 20 different runs with same seed.
Protocol Standard	IEEE 802.11b	Amendment of IEEE 802.11 standard specifications that is direct extension of DSSS.
Media Access Method	CSMA/CA	Multiple access method in which nodes avoid collisions.
Packet Size	64 Bytes	Data chunks.
Mobility Model	Random Waypoint Model	Frequently used mobility model for MANET for its simplicity. Nodes move independently with random velocity within defined node speed. ^[30]
Protocol	UDP	User Datagram Protocol. Connectionless Protocol.
Traffic Type	CBR	Constant Bit Rate Data Consumption.

TABLE 6. Least RMSE and MAE values for standard routing protocols for varying single network parameters.

Protocol	Param	Nodes		Terrain		Speed	
		RMSE	MAE	RMSE	MAE	RMSE	MAE
AODV	Delay	0.173	0.1395	0.2274	0.2023	0.2427	0.1177
	PDR	0.1728	0.0964	0.0726	0.0548	0.2445	0.1373
	Throughput	0.0461	0.0379	0.0698	0.0551	0.1535	0.1426
DSDV	Delay	0.0428	0.0387	0.2307	0.156	0.1405	0.1033
	PDR	0.2054	0.1456	0.0289	0.0227	0.0422	0.0307
	Throughput	0.0431	0.0355	0.0433	0.0305	0.155	0.1147
OLSR	Delay	0.1008	0.0811	0.2049	0.1397	0.1484	0.0785
	PDR	0.207	0.1354	0.0806	0.0677	0.0458	0.0361
	Throughput	0.013	0.01	0.0471	0.0344	0.0503	0.0405
DSR	Delay	0.2571	0.1769	0.2036	0.1617	0.0876	0.0681
	PDR	0.245	0.1672	0.0424	0.0372	0.0342	0.028
	Throughput	0.0731	0.0517	0.0623	0.0457	0.1271	0.0923
AERSA ODV	Delay	0.0881	0.068	0.2633	0.1937	0.2178	0.0877
	PDR	0.1198	0.0843	0.0618	0.0495	0.1892	0.1466
	Throughput	0.0231	0.0175	0.063	0.0491	0.1672	0.1322

be expressed mathematically as shown in (15).

$$RMSE = \left[\left(\frac{1}{k} \right) \sum_{i=1}^k (x_i' - x_i)^2 \right]^{0.5} \quad (15)$$

Hence RMSE is the quadratic scoring rule that measures the square root of squared differences between the estimated data x_i' versus observed data x_i . RMSE is a good estimator of the standard deviation of the distribution of the errors.

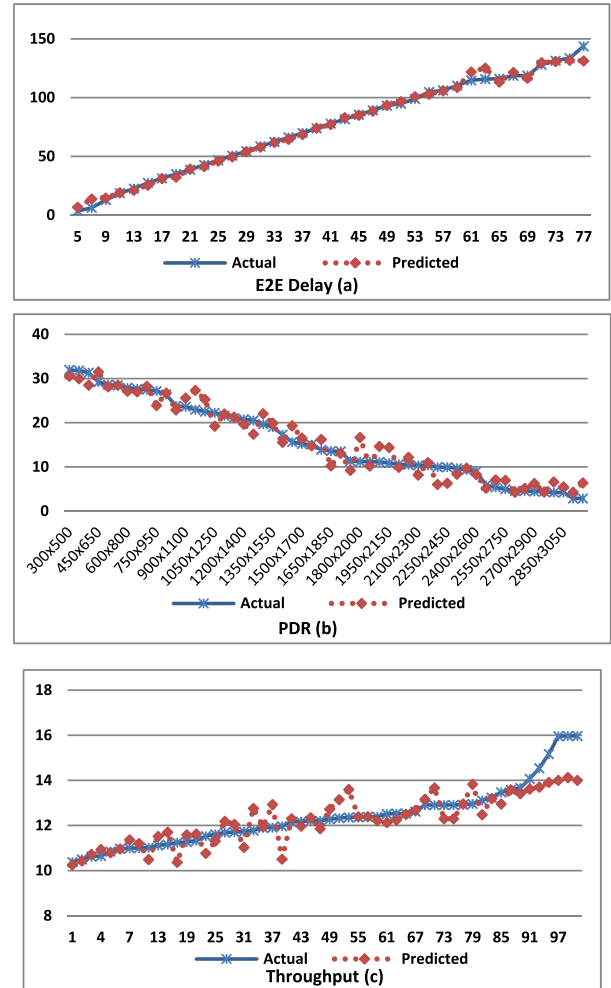


FIGURE 10. Actual versus predicted data for (a) varying network densities by LR and SVM, (b) varying network terrains by KNN and (c) varying max node speeds by MLP in single input feature scenario.

TABLE 7. Minimum RMSE and MAE values for standard routing protocols by varying multiple network parameters.

Protocol	Parameter	RMSE	MAE	ML Algorithm
AODV	Delay	0.2769	0.21	DT
	PDR	0.1958	0.123	SVM
	Throughput	0.1281	0.09	LR
DSDV	Delay	0.1999	0.138	KNN
	PDR	0.1881	0.1135	KNN
	Throughput	0.1019	0.075	KNN
OLSR	Delay	0.1853	0.113	KNN
	PDR	0.1878	0.124	KNN
	Throughput	0.0407	0.071	KNN
DSR	Delay	0.1718	0.116	KNN
	PDR	0.2201	0.138	KNN
	Throughput	0.095	0.067	KNN
AERSAODV	Delay	0.2201	0.113	MLP
	PDR	0.1997	0.123	KNN
	Throughput	0.0982	0.096	KNN

VII. SIMULATION RESULTS AND DISCUSSIONS

This section summarizes the results of the proposed algorithms and compares them with standard protocols for performance evaluation. Table 6 illustrates the best RMSE and MAE attained by the five regression ML algorithms for E2E

TABLE 8. RMSE values for all ML techniques for varying multiple network parameters.

Routing Protocol	Linear Regression			Support Vector Machine			FFANN			KNN			Decision Tree		
	Through put	Delay	PDR	Through put	Delay	PDR	Through put	Delay	PDR	Through put	Delay	PDR	Through put	Delay	PDR
OLSR	0.1334	0.2237	0.2017	0.1412	0.2229	0.195	0.1043	0.1959	0.2241	0.0714	0.1853	0.1878	0.0899	0.2705	0.2624
AODV	0.1281	0.3098	0.1985	0.1302	0.3272	0.1958	0.1325	0.2785	0.2615	0.1505	0.2785	0.2274	0.1375	0.2769	0.2566
DSDV	0.1303	0.2352	0.2123	0.1324	0.2727	0.2346	0.1019	0.2413	0.1924	0.1379	0.1999	0.1881	0.1658	0.2608	0.2506
DSR	0.1202	0.2652	0.2272	0.1232	0.2787	0.2467	0.1139	0.2403	0.2542	0.095	0.1718	0.2201	0.098	0.2866	0.247
V5g	0.1443	0.2836	0.2216	0.145	0.2956	0.2289	0.1537	0.2602	0.2382	0.1545	0.3681	0.1997	0.1466	0.3078	0.1997
V4g	0.1329	0.2787	0.2154	0.1376	0.2201	0.2172	0.1532	0.2273	0.2216	0.1568	0.2956	0.2242	0.1418	0.2313	0.2439
V3g	0.156	0.2213	0.2243	0.157	0.2314	0.2322	0.1384	0.2435	0.238	0.0958	0.2313	0.2213	0.1531	0.2702	0.2531

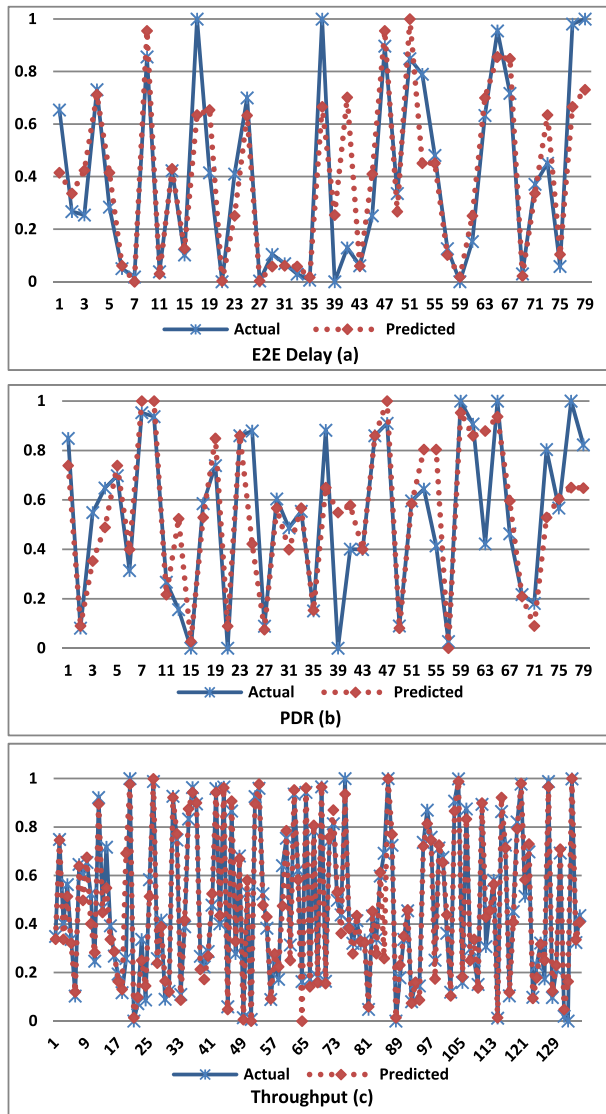


FIGURE 11. (a) Actual versus predicted data of E2E Delay for DSR by KNN, (b) PDR for OLSR by KNN and (c) Throughput for OLSR by KNN in multiple input features scenario while varying multiple network topologies.

delay, PDR and throughput in varying network topology in terms of single predictor variable i.e. network density, terrain size and max nodes speed. It has been evaluated from the results that the FF-ANN/MLP method attained best results for estimations in varying network topologies as compared to other ML algorithms exploited in this work. For throughput

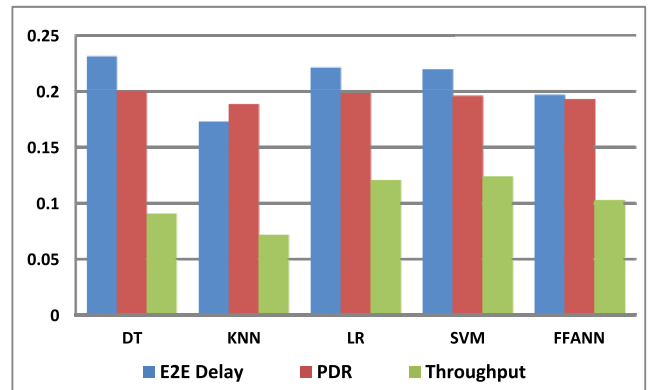


FIGURE 12. Performance of ML techniques in terms of minimum RMSE exploited with multiple input features.

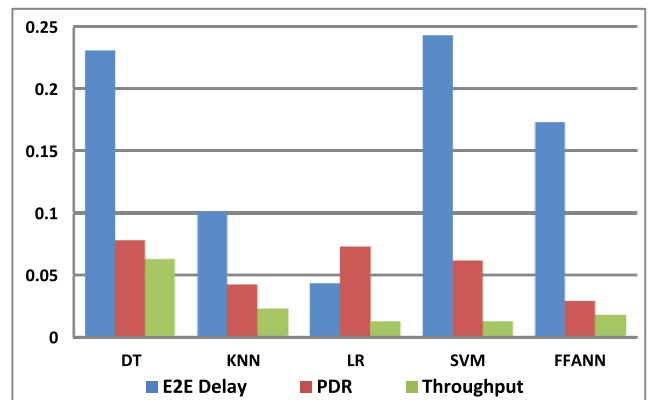


FIGURE 13. Performance of ML techniques in terms of minimum RMSE exploited with single input feature.

prediction, LR and SVM attained RMSE as low as **0.013** and MAE as low as **0.01** in varying network density, KNN has attained lowest RMSE and MAE of **0.0433** and **0.0305** respectively in varying terrain and MLP attained best RMSE and MAE of **0.0503** and **0.0405** respectively in varying nodes speed scenario.

For E2E delay estimation LR attained best RMSE and MAE of **0.0428** and **0.0387** respectively in varying network densities, MLP attained lowest RMSE and MAE in varying nodes speed scenario i.e. **0.0876** and **0.0681** respectively and in varying terrains RMSE and MAE of **0.2036** and **0.1617** are attained by KNN. For PDR prediction MLP attained lowest RMSE and MAE values of **0.0342** and **0.028** in varying

TABLE 9. MAE values for all ML techniques by varying multiple network parameters.

Routing Protocol	Linear Regression			Support Vector Machine			FFANN			KNN			Decision Tree		
	Through put	Delay	PDR	Through put	Delay	PDR	Through put	Delay	PDR	Through put	Delay	PDR	Through put	Delay	PDR
OLSR	0.096	0.1692	0.1415	0.0995	0.1527	0.1423	0.0759	0.145	0.1545	0.0407	0.1132	0.1244	0.065	0.1986	0.2011
AODV	0.0903	0.238	0.1324	0.0924	0.2154	0.1232	0.0939	0.2163	0.171	0.0924	0.2163	0.1503	0.0953	0.21	0.1768
DSDV	0.0918	0.1934	0.1693	0.0912	0.1977	0.1727	0.1436	0.1939	0.1181	0.075	0.1375	0.1135	0.115	0.1998	0.1939
DSR	0.0869	0.2123	0.1573	0.0905	0.2234	0.1502	0.0866	0.1702	0.1831	0.0672	0.1159	0.1376	0.0768	0.2259	0.1838
V5g	0.1043	0.2358	0.1878	0.1042	0.2303	0.1865	0.1188	0.2034	0.1904	0.994	0.1902	0.1495	0.108	0.2418	0.1495
V4g	0.0982	0.1527	0.1424	0.1009	0.1968	0.1789	0.1133	0.1892	0.1756	0.0985	0.2034	0.1231	0.103	0.2432	0.2036
V3g	0.1161	0.1954	0.1234	0.1063	0.1894	0.1812	0.998	0.1132	0.1821	0.1523	0.1939	0.1321	0.1121	0.1892	0.1913

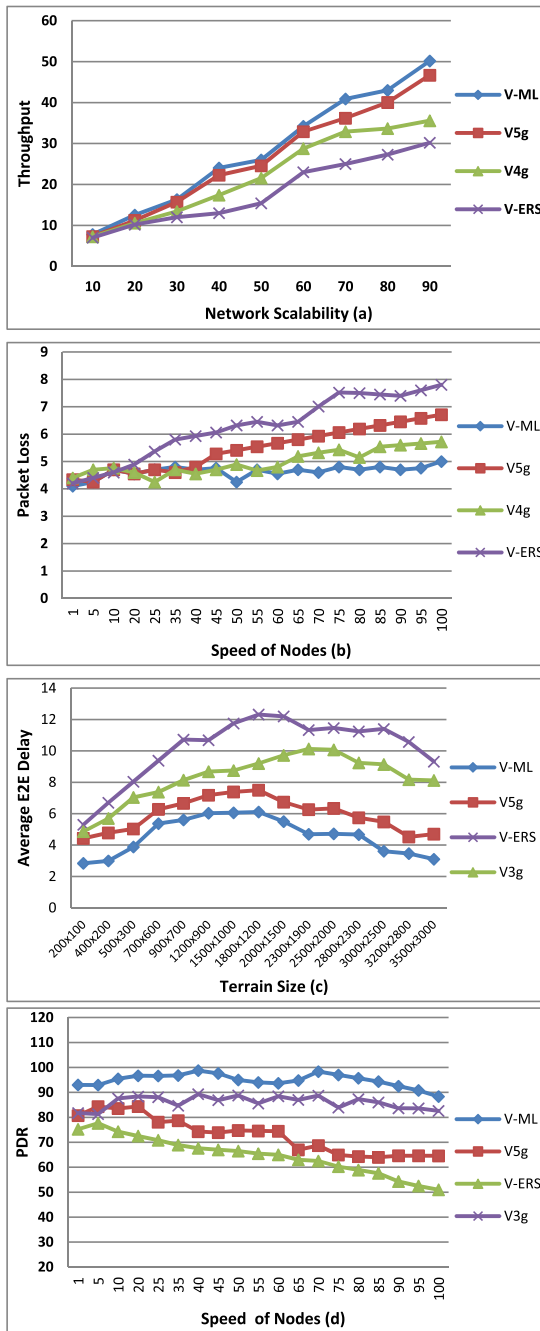


FIGURE 14. Improvement in (a) Throughput, (b) Packet Loss, (c) E2E Delay and (d) PDR for varying network topologies with ML deployment.

nodes speed scenario, MLP attained RMSE and MAE of **0.1198** and **0.0843** respectively in varying network densities,

MLP attained RMSE and MAE as low as **0.0289** and **0.0227** respectively in varying network terrain. Our overall results propose the use of MLP for estimations in most network topologies in order to attain best results. LR algorithm also performs well for estimations in varying network densities. The performance of other techniques, particularly DT, is not as good in most scenarios. Fig. 10 shows the estimations by ML techniques exploited versus actual values used for testing for (a) E2E attained in varying network densities, (b) PDR in varying terrain and (c) throughput in varying speed. It can be observed that the ML techniques have attained much closer estimations to actual routing data.

Table 7 provides minimum RMSE and MAE for standard routing protocols by varying multiple network parameters. It can be illustrated from Table 7 that least RMSE and MAE for throughput, PDR and E2E delay prediction is attained by KNN in most scenarios. These results proposed the use of KNN in multiple input features cases. The least RMSE and MAE attained for E2E delay prediction are **0.1718** and **0.116** respectively, these values for PDR prediction are **0.1878** and **0.124** and for throughput prediction are **0.0407** and **0.071**. Overall performance of the ML algorithms is good for throughput prediction and worst for E2E delay prediction. This implies further refinement and preprocessing of the input data to improve the performance of ML techniques. Fig. 11 provides graphical illustration of precise estimations of exploited ML techniques for E2E delay (a), PDR (b) and throughput in multiple features extracted for training the ML algorithms. The predictions for E2E delay are not as good as that of PDR and throughput. It can be observed that the predictions in case of multiple input features are not as close to actual values as in single input features scenarios. The E2E delay predictions are shown to have not attained closed values due to comparatively high RMSE. This implies improvement in input data for training and testing.

Fig 12 and Fig. 13 provide graphical illustration of performance of all ML techniques exploited in this work in terms of least RMSE attained for predicting E2E delay, PDR and throughput in single and multiple input features respectively. It can be observed in Fig. 12 that KNN attains least error for all estimations whereas DT has comparatively low performance in multi feature scenario. The performance FFANN is also comparatively better while SVM and LR have similar performance. Fig. 13 illustrates that LR has significantly better estimation for E2E delay in single input feature scenario while FFANN outperforms all in estimating the PDR. SVM

and LR have better performance for throughput prediction in single input feature scenario.

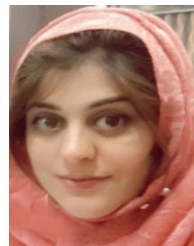
VIII. CONCLUSION AND FUTURE ENHANCEMENTS

Routing protocols in mobile ad hoc networks (MANETs) employ various parametric arrangements in order to attain optimal performance and improve QoS. Researchers are keen to improve and tune these parameters for performance enhancement of the routing protocols in critical scenarios such as link level and node level congestion due to flooding and packets loss. We propose the utilization of machine learning (ML) techniques to enhance the network response in various topological arrangements by exploiting the parametric setup of expanding ring search (ERS) mechanism with random early detection (RED) technique. The comporment of ERS mechanism for handling packets is monitored in terms of hop count, buffer utilization and management at transport level and retransmission of packets at network level. More congestion management techniques are explored from RED technique based on the thresholds and packets dropping criteria that involves levels and rate of packets drop. These mechanisms are monitored and various versions are created by making changes with respect to network topologies for training the ML techniques. The ML techniques are then employed to select the suitable parametric setup among the versions of routing protocols in critical topological arrangement in order to avoid packets loss and congestion. It is shown that the ML techniques particularly KNN and MLP attained high accuracy and low RMSE in predicting the E2E delay, throughput and PDR both in single and multiple in features scenarios. We have evaluated the six variants of standard routing protocols with defined configurations and tested their performance in order to attain improved QoS. The results verify to improve the performance by selection of optimal variants and likewise optimal parametric arrangements. There is supplementary research pull in this work as we aim to apply classification and heuristic methods on further automating the parametric arrangements of various routing mechanisms such as weighted RED, adaptive RED and other variants of RED and ERS based on the significant simulations carried out in this work. These algorithms can be applied on more routing protocols, situations that are congestion prone and more application groups such as IoT, WSN and IETF can be indulged to attain context aware routing.

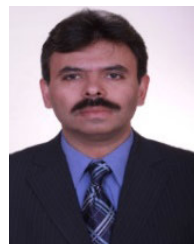
REFERENCES

- [1] T. T. Nguyen, "Multihop routing protocols on wireless ad hoc sensor networks with modified MAC layer and broadcasting scheme," *Concurrency Comput., Pract. Exper.*, vol. 33, no. 2, p. e5720, Jan. 2021.
- [2] P. P. Ray, "A survey on Internet of Things architectures," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 30, no. 3, pp. 291–319, 2018.
- [3] L. Liu, C. Chen, Q. Pei, S. Maharjan, and Y. Zhang, "Vehicular edge computing and networking: A survey," *Mobile Netw. Appl.*, vol. 25, no. 6, pp. 1–24, Jul. 2020.
- [4] N. A. M. Saudi, M. A. Arshad, A. G. Buja, A. F. A. Fadzil, and R. M. Saidi, "Mobile ad-hoc network (MANET) routing protocols: A performance assessment," in *Proc. 3rd Int. Conf. Comput., Math. Statist. (iCMS)*. Singapore: Springer, 2019, pp. 53–59.
- [5] Z. Guo, S. Sheikh, C. Al-Najjar, H. Kim, and B. Malakooti, "Mobile ad hoc network proactive routing with delay prediction using neural network," *Wireless Netw.*, vol. 16, no. 6, pp. 1601–1620, Aug. 2010.
- [6] E. Ghadimi, A. Khonsari, A. Diyanat, M. Farmani, and N. Yazdani, "An analytical model of delay in multi-hop wireless ad hoc networks," *Wireless Netw.*, vol. 17, no. 7, pp. 1679–1697, Oct. 2011.
- [7] N. Javaid, Z. A. Khan, U. Qasim, M. Jamil, M. Ishfaq, and T. A. Alghamdi, "Modeling routing overhead of reactive protocols at link layer and network layer in wireless multihop networks," *Math. Problems Eng.*, vol. 2015, pp. 1–14, Jul. 2015.
- [8] Y. Mai, F. M. Rodriguez, and N. Wang, "CC-ADOV: An effective multiple paths congestion control AODV," in *Proc. IEEE 8th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Las Vegas, NV, USA, Jan. 2018, pp. 1000–1004.
- [9] M. H. Malik, A. Majeed, M. E. Aydin, and M. H. Malik, "A parametric study for congestion control in queuing networks," in *Proc. Int. Conf. Future Netw. Distrib. Syst.*, New York, NY, USA, Jul. 2017, pp. 1–5.
- [10] K.-Y. Gwak, S. Park, and J. Song, "Performance evaluation: Two flood-cancellation methods of the blocking expanding ring searches on the AODV/WiFi MANET environment," in *Proc. 21st Int. Conf. Adv. Commun. Technol. (ICACT)*, PyeongChang, South Korea, Feb. 2019, pp. 239–248.
- [11] A. H. Shintre and S. Sondur, "Improved blocking expanding ring search (I-BERS) protocol for energy efficient routing in MANET," in *Proc. Int. Conf. Recent Adv. Innov. Eng. (ICRAIE)*, Jaipur, India, May 2014, pp. 1–6.
- [12] L. Shibao and J. Wei, "AODV route protocol research based on improved ERS algorithm," in *Proc. IEEE Int. Conf. Wireless Commun., Netw. Inf. Secur., Beijing, China, Jun. 2010*, pp. 102–106.
- [13] K. U. Adhvaryu, "Energy efficient ERS for multicast routing in MANET," in *Proc. Int. Conf. Recent Adv. Energy-Efficient Comput. Commun. (ICRAECC)*, Nagercoil, India, Mar. 2019, pp. 1–4.
- [14] I. Topalova and P. Radoyska, "Control of traffic congestion with weighted random early detection and neural network implementation," in *Proc. 14th Int. Conf. Autom. Auto. Syst. (ICAS)*, Nice, France, 2018, pp. 8–12.
- [15] S. Sharma, D. Jindal, and R. Agarwal, "Analysing mobile random early detection for congestion control in mobile ad-hoc network," *Int. J. Electr. Comput. Eng.*, vol. 8, no. 3, pp. 1305–1314, 2018.
- [16] A. Bujari, A. Marin, C. E. Palazzi, and S. Rossi, "Analysis of ECN/RED and SAP-LAW with simultaneous TCP and UDP traffic," *Comput. Netw.*, vol. 108, pp. 160–170, Oct. 2016.
- [17] B. Siregar, M. S. Manik, R. Rahmat, U. Andayani, and F. Fahmi, "Implementation of network monitoring and packets capturing using random early detection (RED) method," in *Proc. IEEE Int. Conf. Commun., Netw. Satell. (Comnetsat)*, Semarang, Indonesia, Oct. 2017, pp. 42–47.
- [18] N. Sharma, S. S. Rajput, A. K. Dwivedi, and M. Shrivastava, "P-RED: Probability based random early detection algorithm for queue management in MANET," in *Proc. Adv. Comput. Comput. Sci.* Singapore: Springer, 2018, pp. 637–643.
- [19] S. B. Danladi and F. U. Ambursa, "DyRED: An enhanced random early detection based on a new adaptive congestion control," in *Proc. 15th Int. Conf. Electron., Comput. Comput. (ICECCO)*, Abuja, Nigeria, Dec. 2019, pp. 1–5.
- [20] A. Bujari, A. Marin, C. E. Palazzi, and S. Rossi, "Smart-RED: A novel congestion control mechanism for high throughput and low queuing delay," *Wireless Commun. Mobile Comput.*, vol. 2019, pp. 1–10, Apr. 2019.
- [21] C. Perkins, E. Belding-Royer, and S. Das, *Ad Hoc on Demand Distance Vector (AODV) Routing*, document RFC 3561, IETF MANET Working Group, CA, USA, 2003, pp. 1–37.
- [22] P. K. Maurya, G. Sharma, V. Sahu, A. Roberts, M. Srivastava, and M. T. Scholar, "An overview of AODV routing protocol," *Int. J. Mod. Eng. Res.*, vol. 2, no. 3, pp. 728–732, 2012.
- [23] N. Akhtar, M. A. K. Khattak, A. Ullah, and M. Y. Javed, "Congestion aware and adaptive routing protocols for MANETs: A survey," in *Recent Trends and Advances in Wireless and IoT-Enabled Networks*, vol. 1, 1st ed. Cham, Switzerland: Springer, 2019, pp. 159–169.
- [24] J. Zhou, X. Chen, Y. Deng, L. Cui, K. Wei, and G. Zhong, "Maximum weight packet scheduling scheme with retransmission in multi-hop wireless networks," in *Proc. 10th Int. Conf. Mobile Comput. Ubiquitous Netw. (ICMU)*, Toyama, Japan, Oct. 2017, pp. 1–4.
- [25] N. Javaid, A. Bibi, S. N. Mohammad, Z. A. Khan, and N. Alrajeh, "Towards optimizing routing overhead in wireless multi-hop networks," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 19, nos. 1–2, pp. 4–18, 2015.

- [26] T. Niu, D. Zhang, K. Xu, L. Wu, and Y. Jiang, "Buffer-aided instantly decodable network coding for wireless video broadcasting with unequal packet-priority retransmission," in *Proc. 16th Int. Symp. Commun. Inf. Technol. (ISCIT)*, Qingdao, China, Sep. 2016, pp. 117–122.
- [27] Z. Guo and B. Malakooti, "Delay prediction for intelligent routing in wireless networks using neural networks," in *Proc. IEEE Int. Conf. Netw., Sens. Control*, Fort Lauderdale, FL, USA, 2006, pp. 625–630.
- [28] A. Ghaffari, "Real-time routing algorithm for mobile ad hoc networks using reinforcement learning and heuristic algorithms," *Wireless Netw.*, vol. 23, no. 3, pp. 703–714, Apr. 2017.
- [29] H. Yang, Z. Li, and Z. Liu, "A method of routing optimization using CHNN in MANET," *J. Ambient Intell. Humanized Comput.*, vol. 10, no. 5, pp. 1759–1768, May 2019.
- [30] W. Su, S. J. Lee, and M. Gerla, "Mobility prediction and routing in ad hoc wireless networks," *Int. J. Netw. Manage.*, vol. 11, no. 1, pp. 3–30, 2001.
- [31] L. Hongyan, W. Hong, and G. Chao, "Internet time-delay prediction based on autoregressive and neural network model," in *Proc. Int. Conf. Commun., Circuits Syst.*, Guilin, China, Jun. 2006, pp. 1758–1761.
- [32] S. R. S. Tabib and A. A. Jalali, "Modelling and prediction of Internet time-delay by feed-forward multi-layer perceptron neural network," in *Proc. 10th Int. Conf. Comput. Modeling Simulation (uksim)*, Cambridge, U.K., 2008, pp. 611–617.
- [33] H. Sun and H. D. Hughes, "Adaptive QoS routing based on prediction of local performance in ad hoc networks," in *Proc. IEEE Wireless Commun. Netw. (WCNC)*, New Orleans, LA, USA, vol. 2, Mar. 2003, pp. 1191–1195.
- [34] D. Nayab, M. H. Zafar, and M. Basher, "Adaptive expanding ring search based per hop behavior rendition of routing in MANETs," *Comput., Mater. Continua*, vol. 67, no. 1, pp. 1137–1152, 2021.
- [35] I. Diakonikolas, W. Kong, and A. Stewart, "Efficient algorithms and lower bounds for robust linear regression," in *Proc. 13th Annu. ACM-SIAM Symp. Discrete Algorithms, Soc. Ind. Appl. Math.*, San Diego, CA, USA, 2019, pp. 2745–2754.
- [36] A. Karpus, M. Raczynska, and A. Przybylek, "Things you might not know about the K-nearest neighbors algorithm," in *Proc. 11th Int. Joint Conf. Knowl. Discovery, Knowl. Eng. Knowl. Manage.*, Vienna, Austria, 2019, pp. 539–547.
- [37] C. Eypoglu, "Breast cancer classification using K-nearest neighbors algorithm," *Online J. Sci. Technol.*, vol. 8, no. 3, pp. 29–34, 2018.
- [38] S. A. Ludwig, S. Picek, and D. Jakobovic, "Classification of cancer data: Analyzing gene expression data using a fuzzy decision tree algorithm," in *Proc. Oper. Res. Appl. Health Care Manage.* Cham, Switzerland: Springer, 2018, pp. 327–347.
- [39] H. Zhao and X. Li, "A cost sensitive decision tree algorithm based on weighted class distribution with batch deleting attribute mechanism," *Inf. Sci.*, vol. 378, pp. 303–316, Feb. 2017.
- [40] I. Ahmad, M. Basher, M. J. Iqbal, and A. Rahim, "Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection," *IEEE Access*, vol. 6, pp. 33789–33795, 2018.
- [41] A. Anzer and M. Elhadef, "A multilayer perceptron-based distributed intrusion detection system for Internet of vehicles," in *Proc. IEEE 4th Int. Conf. Collaboration Internet Comput. (CIC)*, Philadelphia, PA, USA, Oct. 2018, pp. 438–445.
- [42] M. Duraipandian, "Performance evaluation of routing algorithm for MANET based on the machine learning techniques," *J. Trends Comput. Sci. Smart Technol.*, vol. 1, no. 1, pp. 25–38, Sep. 2019.
- [43] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, and J. Vanderplas, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.
- [44] L. Maygaa-Marcillo, L. Urquiza-Aguilar, and M. Paredes-Paredes, "Creating a new model in NS3 network simulator," in *Preprints*. Basel, Switzerland: MDPI, 2019, pp. 1–5.
- [45] A. S. Toor and A. K. Jain, "A survey on wireless network simulators," *Bull. Electr. Eng. Informat.*, vol. 6, no. 1, pp. 62–69, Mar. 2017.
- [46] K. Adhvaryu, "Performance comparison of multicast routing protocols based on route discovery process for MANET," in *Inventive Communication and Computational Technologies*, vol. 89, 1st ed. Singapore: Springer, 2020, pp. 79–85.
- [47] N. E. Majd, N. Ho, T. Nguyen, and J. Stolmeier, "Evaluation of parameters affecting the performance of routing protocols in mobile ad hoc networks (MANETs) with a focus on energy efficiency," in *Proc. Future Inf. Commun. Conf.*, vol. 70, 1st ed. Cham, Switzerland: Springer, 2019, pp. 1210–1219.
- [48] G. Raina, S. Manjunath, S. Prasad, and K. Giridhar, "Stability and performance analysis of compound TCP with REM and drop-tail queue management," *IEEE/ACM Trans. Netw.*, vol. 24, no. 4, pp. 1961–1974, Aug. 2016.
- [49] T. Xing, Y. Liu, B. Tang, and F. Wu, "Dynamic-adjusting AODV routing protocol based on max hop count," in *Proc. IEEE Int. Conf. Wireless Commun., Netw. Inf. Secur.*, Beijing, China, Jun. 2010, pp. 535–539.
- [50] T. Chai and R. R. Draxler, "Root mean square error (RMSE) or mean absolute error (MAE)?—Arguments against avoiding RMSE in the literature," *Geoscientific Model Develop.*, vol. 7, no. 3, pp. 1247–1250, Jun. 2014.
- [51] D. Krishnamoorthy, P. Vaiyapuri, A. Ayyanar, Y. Harold Robinson, R. Kumar, H. V. Long, and L. H. Son, "An effective congestion control scheme for MANET with relative traffic link matrix routing," *Ara-bian J. Sci. Eng.*, vol. 45, no. 8, pp. 6171–6181, Aug. 2020.



DURR-E-NAYAB is currently pursuing the Ph.D. degree in machine learning techniques for their applications in mobile ad hoc networks with the University of Engineering and Technology (UET), Peshawar, Pakistan. She has served as a Research Assistant with the Center for Intelligent Networks and Research (CISNR), UET, Peshawar, where she is also working in collaboration with various international research projects. She is also serving as a Lecturer with UET. Her research interests include mobile ad hoc networks, routing schemes in wireless networks, intelligent systems design, artificial intelligence applications, and machine learning algorithms.



MOHAMMAD HASEEB ZAFAR (Senior Member, IEEE) received the Ph.D. degree in electronic and electrical engineering (EEE) from the University of Strathclyde, in 2009. He is currently a Professor with the Faculty of Electrical and Computer Engineering, University of Engineering and Technology, Peshawar, Pakistan. He is also a Professor with the Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia. His main research interests include performance analysis of diverse computer and wireless communication networks and systems. He is a Lifetime Member of Pakistan Engineering Council.



ALI ALTALBE received the M.Sc. degree in information technology from Flinders University, Adelaide, SA, Australia, and the Ph.D. degree in information technology from The University of Queensland, Brisbane, QLD, Australia. He is currently working as an Assistant Professor with the Department of IT, King Abdulaziz University, Jeddah, Saudi Arabia. His current research interests include measuring information systems effectiveness, mobile services, e-learning, and human–computer interaction (HCI).

...