

# Improving QoS of DYMO Routing Protocol in MANETs using Random Forest Classifier Technique

Jagdeep Kaur<sup>1</sup>, Ankita<sup>1</sup>

<sup>1</sup> CT University, Ferozpur Road Ludhiana, Punjab, India

---

**Article History:**

**Received:** 12-12-2024

**Revised:** 25-01-2025

**Accepted:** 05-02-2025

**Abstract:**

A network known as mobile ad hoc network (MANET) is made up of several wireless mobile nodes that connect with one another without the need of constitutional administration or network infrastructure. The mobile nodes have the ability to act as both the host and a router, sending packets from the source node to the destinations. Routing in MANETs is difficult since mobile nodes are constantly moving, the topology is always changing, and occasionally routes fail, resulting in a decline in network performance. Random Forest Classifier machine learning is an effective procedure for modifying productive routing protocols for MANETs. Aforementioned paper illustrates the characteristics and working of DYMO routing protocol. Further we implemented Random Forest Classifier technique on DYMO protocol using Google Collaboratory simulator and correlated its simulator results with elementary DYMO protocol.

**Keywords:** MANETs, DYMO, Random Forest classifier.

---

## 1. Introduction

A Mobile ad hoc Networks (MANETs) is a wireless ad-hoc networks. The name "ad hoc networks" has Latin origins and means "for this" denoting anything improvised or created on the spot. MANETs is created spontaneously, just as laptops and other devices like phones and tablets. It frequently doesn't make use of the more sophisticated network protocols and services that are present on networks that have been established by an organization. It does not have a central point; thus, users can connect at any time. This term refers to all devices, also known as nodes, that have the ability to send and receive data [4]. When traditional connectivity was scarce or non-existent, MANETs were used to handle emergency situations that arose in those locations or during those periods. The nodes are temporarily connected to one another in order to use the wireless medium for ad hoc purposes, such as providing connectivity and services when and wherever needed, and then break off communication [3].

MANETs can be used without first requiring setup. It is not required to be fixed in a single location. In MANET, there is no use of routers and access points. The nodes have unrestricted mobility. The hardest aspect is actually building the MANET. It is difficult to make sure every device has access to the data it needs to properly guide traffic. These networks might be connected to or distinct from the larger internet. There might be one or more transceivers all of which are unique between two nodes. Consequently, the topology is very self-sufficient and is subject to frequent modifications. It might work better for someone who wants to quickly set up a wireless network with a small number of devices, but not for someone who wants to build up a network that will last for a long time and include

many devices. They have the advantage of not requiring setup and being swiftly assembled in the event of an emergency, such as a natural disaster or armed conflict [4].

Because MANETs are dynamic and decentralized, they provide special routing challenges. Because of factors including node mobility, low bandwidth, and inconsistent communications links, traditional routing protocols designed for wired networks are occasionally inadequate for MANETs. In order to address these problems and enhance communication between mobile nodes, specific routing protocols have been developed [1].

Here DYMO routing protocol of MANET and machine learning techniques are center of attention. Because of the movable nodes, dispersed control, and dynamic topology of MANETs, routing is a difficult problem. MANET's routing protocol governs how mobile nodes choose which path to take while routing packets between other nodes [17]. The goal of current MANETs routing protocols research is to identify the most effective path between mobile nodes. Three types of routing protocols are distinguished: Reactive, Proactive, Hybrid routing protocols [17][3]. The main goal of this paper gives the review of working of DYMO routing protocol and machine learning techniques.

This paper is organized as follow. In section II describes the classification of routing protocols in MANET. The section III and IV explains the working of DYMO routing protocol, section V describes the Random Forest classifier machine learning technique. Finally, V, VI and VII sections explain the simulation environment, Experimental results, summary and future perspective of the paper.

## **2. Routing Protocols**

The MANET routing mechanism is in charge of locating, creating, and keeping up a route between two mobile nodes. There are two paradigms available for packet routing: single-hop and multi-hop. The destination node is considered to be within the source node's communication range under the one hop paradigm. As a result, the source node and destination can establish a direct connection. In the multi-hop paradigm, when the destination is out of the source node's communication range, the source node can still communicate with it through intermediary nodes. Known as a multi-hop network, MANETs are made up of mobile nodes that cooperate to relay control or data packets from one source node to another.

The mobile nodes are engaged in route discovery, and the intermediate mobile nodes along the routes play crucial roles in route maintenance once they are located. A few challenges in creating a route between source and destination nodes via intermediary nodes are low bandwidth, restricted coverage and connectivity because of the transmission range limitation, increased error rate, increased likelihood of interference, power consumption, lack of a centralized routing mechanism, and frequent changes in network topology because of mobility. In MANET routing becomes a more intricate operation [2].

Route tables are pre-established and are maintained continuously over the course of the network's lifetime by the routing protocols used in their deployment and operation on wired networks. Yet, in an environment that is changing quickly, existing routing protocols are inefficient or ineffective at sustaining end-to-end connectivity and delivering network services. As a result, new routing protocols were created to meet the unique requirements of MANETs [3].

These routing protocols were classified as either proactive or reactive, for example, AODV (Ad-hoc On-Demand Distance Vector), DYMO (Dynamic MANET On-Demand), DSR (Dynamic Source Routing), TORA (Temporary Ordered Routing Algorithm), DSDV (Destination Sequenced Distance vector), OLSR (Optimized Link State Routing), FSR (Fisheye State Routing), and CGSR (Cluster Head Gateway Switch Routing). However, in some implementations, concepts from both categories were used, resulting in hybrid solutions, such as ZRP (Zone Routing Protocol) and GRP (Gathering-based Routing Protocol). Whereas in reactive routing protocols a node's route is produced only, when necessary, in proactive routing protocols every node creates and manage a routing table along with the routes to all other common nodes in the network [3].

### 3. DYMO Routing Protocol

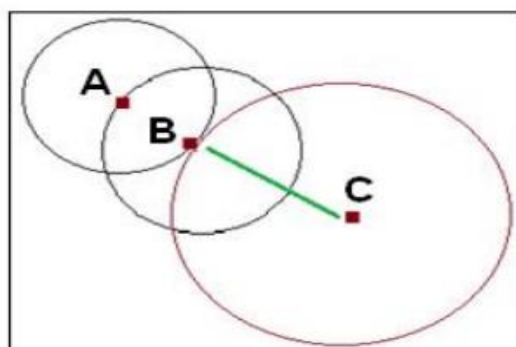
Dynamic Manet On-demand (DYMO) routing protocol is a reactive routing protocol created for MANETs [17]. It is a multihop network routing protocol that is easy to use and quick [18]. The DYMO routing protocol, commonly known as AODVv2, is a continuation of the AODV routing system and shares many of its features [11]. DYMO is rationalize merging of the AODV and DSR routing protocols. The fundamental features of the route maintenance and discovery phases are retained, and it functions similarly to AODV. By adding own address to the its data packet, each node that forwards an RREQ or RREP in DSR functions in this way [10].

The two protocol procedures that make up DYMO are route maintenance and discovery. Route discovery is carried out when an AODVv2 router has to send a packet to a location for which it does not have a path. Routine maintenance is carried out to prevent dropping packets when an active route fails and to prevent prematurely removing routes from the route table [11]. Nodes keep a routing table updated with data gathered during the discovery phase. The node's updated position in the network topology is identified by a sequence number that is labelled on the routing table. During the route discovery process, a node updates its position by incrementing its sequence number in its own database when it creates a routing message. A routing table has a list of target addresses connected to the target routing table's sequence number and metric. The DYMO metric, known as hop count, counts the bare minimum of nodes a packet must travel through in order to reach its destination. The route with the lowest hop count value is the best one. Nodes only make updates to their routing tables when they receive data from nodes whose string number is greater than or equal to the one that is cached. Sequence numbers are used in this manner to prevent outdated data from being updated in the table [18].

#### *A. Unique attributes of the DYMO Protocol*

##### *1) Locality Perception*

Link failures can be found using the MANET Neighborhood Discovery Protocol (NHDP), route timeouts, link layer feedback, and HELLO messages. However, to ensure that the neighboring neighbors are still active, the DYMO protocol does not employ any form of HELLO message.



**Figure 1: Locality Perception**

DYMO protocol has an intriguing characteristic in that it makes use of feedback from the lower layer to maintain locality. In this kind of situation, the lower layer protocol feedback could be helpful, as seen in Figure 1. Node A can reach node B in this instance. Node A's link layer uses a carrier sense approach to determine whether the channel is free for transmission when node B is broadcasting to node C. The fact that node B is still operational and transmitting can be inferred from this as feedback.

## 2) *Path Accumulation*

The goal of route discovery is to catch a path towards destination router by dispatching RREQ messages across the network from the source router. As soon as an intermediate router receives an RREQ, recording and rebroadcasting a route and RREQ to the originator and done together with its own data, known as the path accumulation function. Sending an RREP to the originator, the destination's router collect the RREQ. Route establishment occurs when the originator receives the RREP. DYMO and AODV have comparable route maintenance. DYMO's path accumulation function incorporates source routing attributes, which permits the nodes that are interested for routing messages to learn about routes to other nodes without having to start their own route request discoveries. Although the routing packet's size has increased, this path accumulation service can thereby lower routing overhead [11].

## *B. Route Messages*

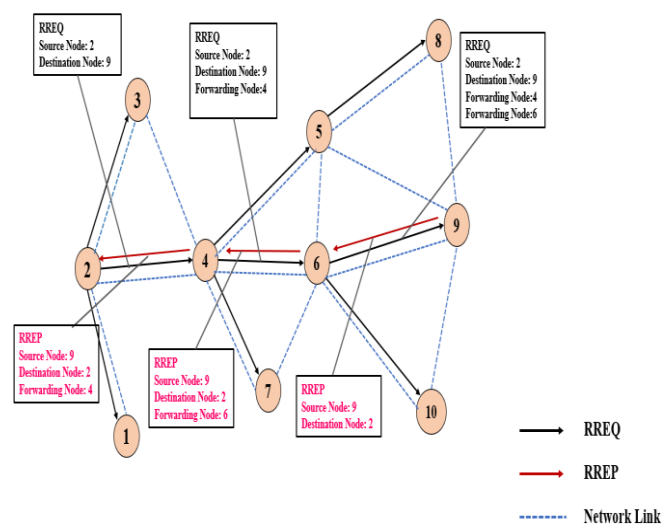
Route Request (RREQ), Route Reply (RREP), and Route Error (RERR) are the three messages that DYMO implements during the routing procedure.

- 1) The source node uses the RREQ message trying to find a legitimate path to a certain destination node.
- 2) A route between the source and destination nodes, as well as all of the intermediate nodes in between, is established using the RREP message.
- 3) The baseless route from any intermediate node to the destination node is indicated by the RERR message.

In addition, the DYMO protocol requires every node to keep a unique, unsigned integer known as a "sequence number," which ensures that packets are delivered to their destination in an ordered manner and preserves loop-free routes that resemble those in DSDV and AODV. Sequence numbers enable the nodes to assess how current the routing data is [13].

### C. Route Discovery

Using Figure 2 as an example, we demonstrate route discovery method. Since node 2 in the figure wishes to connect with node 9 and node 2 is represented as source node while node 9 is represented as the destination node. The node 2 has its own sequence number which is increased before it is incorporated to the RREQ, as well as its own address in the RREQ message. At last, a hop count with a value of 1 is added for the source. The destination node 9 is then further described. The address of the destination is the most crucial component. A node forwards an RREQ only if it hasn't already executed. This allows for a controlled spread of messages throughout the network. In order to identify this, the sequence number is used [10].



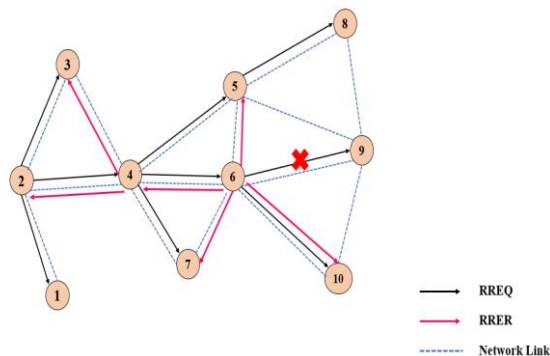
**Figure 2: Route Discovery Process**

Just like the source node, any node forwarding an RREQ may attach to the RREQ its own address, sequence number, prefix, and gateway information. When the destination sends back an RREP message, the source node will wait to receive it. Another RREQ may be issued by the node in an attempt to find a route if no RREP is received during the RREQ WAIT TIME. As stated in the DYMO specification, the default value of the constant RREQ WAIT TIME is 1000 milliseconds. When nodes 4 and 6 propagate the RREQ from node 2, they append information to it, as seen in figure 2. Upon receiving an RREQ, a node examines the addresses and related data contained within the message. The information about node 9, including its address, sequence number, prefix, and gateway, is then included in an RREP message that is constructed in response to the RREQ. The RREP message is then transmitted back via unicast down the reverse path. Asymmetric linkages are not supported by DYMO since replies are transmitted on the opposite path. Nodes forwarding an RREQ do the same packet processing as nodes forwarding an RREP; that is, forward routes to nodes that have added their address block to the RREP can be created using the information included in the RREP [14].

### C. Route Maintenance

Route Maintenance is the mechanism of reacting towards topology change that occur after a route is first established. Nodes constantly check the active links to maintain pathways, and as they receive

and send data packets, they amend the Valid Timeout field of the entries in their routing table. A node must send a Route Error (RERR) message in response to receiving a data packet for a destination for which it does not have a valid route [16]. The address and sequence number of the unreachable node are included in a list that the node creates while generating the RERR message. Furthermore, as the next hop entry, the node adds to the routing table all entries that depend on the unreachable destination. Notifying people about additional routes that are unavailable is the goal. The catalogue is dispatched in the RERR packet by the node. It broadcasts the RERR message.



**Figure 3: Route Maintenance Process**

Figure 9 provides an illustration of the diffusion process. When the connection between nodes 6 and 9 breaks, node 6 gets a data packet related to node 9. One possible explanation for the term "broken link" is that the route table record for a node may have become invalid due to a timeout in the time stamp. An RERR message is generated by node 6 and is sent backwards to node 2. A node comparing the list of nodes in the RERR with the matching entries in its routing table is what happens when it receives an RERR. When the next hop node is the same as the node from whence the RERR was received, and the entry's arrangement number is more than or equal to the sequence number contained in the RERR, a route table entry for that node from the RERR is deemed invalid. The relevant way in the RERR's list of distant nodes needs to be deleted if a route table entry is not overruled. The node stops propagating this RERR if there are no more entries. If not, more broadcasts of the RERR are made. The aforementioned sequence number check is carried out to exclusively invalidate new routes and to stop the spread of outdated information. The RERR distribution aims to notify any node that might be connected to a link in the event of a failure. Since a node can only forward an RERR message once, RERR propagation is guaranteed to end [14].

#### 4. Random Forest Classifier Machine Learning Technique

The Random Forest classifier is made up with a group of tree-structured classifiers called  $\{h(x, \Theta_k) \mid k=1, 2, \dots\}$ . Each and every tree casts a unit vote for the most favoured class at input  $x$ , and the  $\{\Theta_k\}$  are independent, identically distributed random vectors [12].

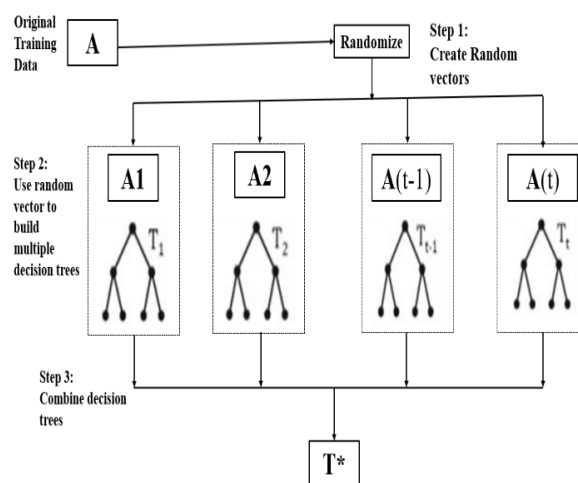
Random forest is an altogether strategy that uses several instances of decision trees, each randomly selected from a different portion of the training set. Bootstrapping samples are what these examples are known as. Generally, all of the bootstrapped samples vote to determine the final results [8].

The Random Forest technique yields a set of decision trees. Breiman chose a randomization strategy that works in a well-behaved manner with bagging or random subspace approaches in order to create

different variations among base decision trees. Breiman used the following procedures to create every single tree in Random Forest: If there are  $N$  records in the training set, then  $N$  records are randomly selected from the authentic data by making use of bootstrap sample, which takes the place of the original records. This sample will serve as the tree's training set. When there are  $M$  input variables, a number  $m \ll M$  is chosen so that, at each node,  $m$  variables are picked at random from  $M$ , and the node is split using the best split on these  $m$  characteristics. The forest grows while keeping the value of  $m$  consistent. Every tree is cultivated as much as it can. No pruning is done.

In this way, a lot of trees are introduced into the forest; the parameter  $N$  tree predetermines the number of trees. Each node's chosen number of variables ( $m$ ). Node size, which is the number of instances in the leaf node, is a parameter that can be used to adjust the tree's depth which is usually set to one. Once the forest has been constructed or trained as previously described, it is run across all of the trees that are grown there in order to classify a new instance. The new instance that is registered as a vote is classified by each tree. The class for which the most votes are counted (majority voting) is proclaimed as the classification of the new instance after combining the votes from all trees [12].

The following graphic, shown in figure 4, provides a visual representation of the Random Forest technique [6].



**Figure 4: The Random Forest approach depicted**

Random forest works best when analysing complex data structures concealed in a simple record with fewer than 10,000 rows but possibly millions of columns [18].

When using the DYMO protocol in Mobile Ad Hoc Networks (MANETs), utilizing Random Forest, a potent ensemble learning technique, can provide several noteworthy advantages:

- **Improved Route Selection:** To improve routing choices, Random Forest can be utilized to analyse several parameters (such as signal strength).
- **Predictive Analytics:** Random Forest can anticipate possible network congestion or connection failures by utilizing past data and patterns. This enables DYMO to proactively reroute traffic and steer clear of troublesome paths.

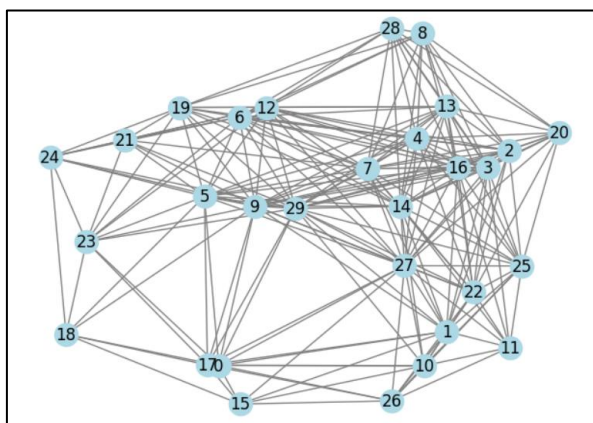
- **Real-Time Adaptation:** Random Forest continuously learns from fresh data, enabling it to quickly adjust to changing network conditions. Due to this, routing decisions are guaranteed to stay optimal even in the extremely dynamic settings found in MANETs.
- **Handling Mobility:** Node mobility patterns can be taken into account by the algorithm, making sure that routes remain stable and are less likely to break as a result of node movement.
- **Fault Tolerance:** Random Forest is able to identify anomalous packet loss or delays, among other anomalies in network behaviour, that may point to malicious activity or connection failures. This makes it possible for DYMO to act quickly to address the issue.
- **Security:** Random Forest can improve network security by aiding DYMO to steer clear of vulnerable nodes by spotting patterns linked to attacks (such as misbehaving routing).

### 5. Simulation Environment

Python language is used as the simulation to analyse the performance of the DYMO routing protocol. The simulation environment discussed the criterion implemented to simulate the MANET environment in the Google Colaboratory simulation tool. Simulation area 100m with fixed simulation time 10s. The number of nodes is 30 nodes. The brief simulation parameters as shown in table 1.

**Table 1: Summary of Simulation Parameters**

Simulation Parameters	Values
Simulation Field	100 sq.m
Simulation tool	Google Colaboratory
Number of nodes	30
Routing protocol	Dynamic MANET On-Demand (DYMO)
Machine Learning Technique	Random Forest Classifier
Maximum Speed	5 units per seconds
Pause time	2 seconds
Simulation time	10 seconds



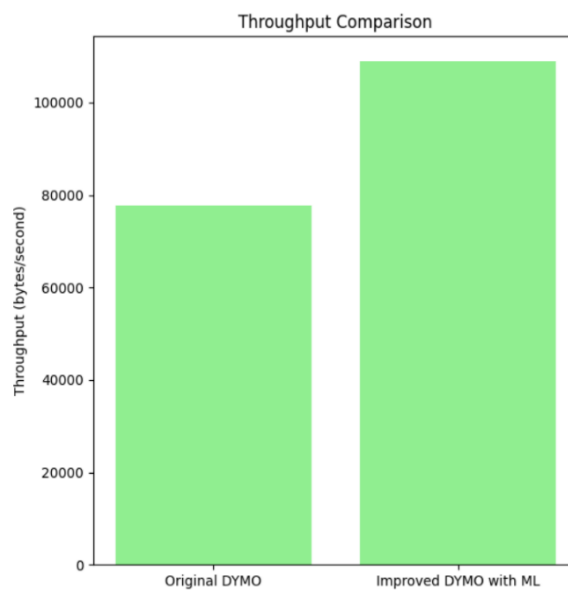
**Figure 5: Simulation Running with 30 nodes**

## 6. Experimental Results

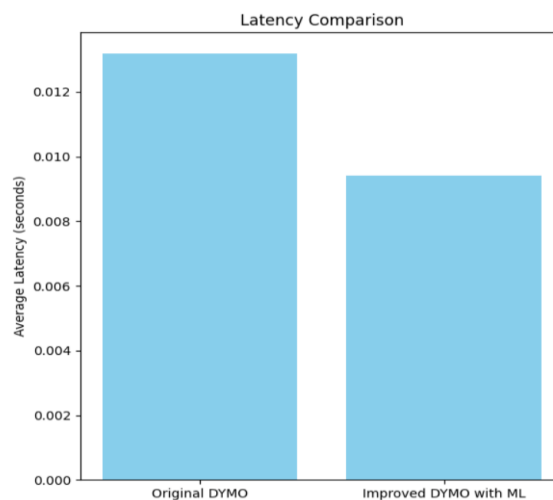
The evaluation of the performances DYMO routing protocol based on Random Forest Classifier analysed and calculated via following performance pattern:

1. Throughput: It is the number of packets that has been successfully delivered per unit time and is referred to the amount of transmission data from source to destination node [10][5].
2. Average Latency: This is the interval time between sending and receiving by the source node and the destination node respectively, which also includes the processing time and queuing time [9].

Figure 6 shows the throughput of original DYMO routing protocol and Random Forest Classifier machine leaning technique based DYMO. It indicates that original DYMO give 77688 bytes/sec and machine learning based DYMO give 108845 bytes/sec.



**Figure 6: Throughput of original DYMO and Random Forest Classifier ML based DYMO**



**Figure 7: Average Latency of original DYMO and Random Forest Classifier ML based DYMO**

Average latency time of original DYMO is 0.013180 seconds and Random Forest Classifier machine learning technique based DYMO is 0.009407 seconds as shown in figure 7. Consequently, the machine learning-based DYMO protocol performs better than the basic DYMO methodology.

## 7. Summary and Future Perspective

In this paper, we have profitably simulated the DYMO based on Random Forest Classifier machine learning technique and analysed the performance using various simulation metrics. By utilizing Random Forest Classifier's capabilities, we can enhance the DYMO protocol's overall efficiency, optimize routing decisions, and predict network circumstances with more accuracy using Python. Machine learning integration makes adaptive and intelligent routing possible, which is essential in dynamic and often unpredictable environments of MANETs. Random forest ML technique also improve Quality of service of the DYMO protocol. It lowers the latency time between the source and the destination and increases throughput. Future work can explore the integration of additional machine learning models and real-world testing to further validate and refine the proposed improvements. The purpose of this paper is to assist researchers working on the DYMO protocol.

### References:

- [1] DR. S. HEMALATHA *et al.*, “ENHANCED ROUTING PROTOCOL IN MOBILE AD HOC NETWORK FOR IMPROVING THE PERFORMANCE,” *Journal of Theoretical and Applied Information Technology*, vol. 102, no. 14, pp. 5386–5395, Jun. 2024, Accessed: Jun. 31, 2024. [Online]. Available: <https://www.jatit.org/>
- [2] Baidaa Hamza Khudayer *et al.*, “A Comparative Performance Evaluation of Routing Protocols for Mobile Ad-hoc Networks,” *International journal of advanced computer science and applications/International journal of advanced computer science & applications*, vol. 14, no. 4, Jan. 2023, doi: <https://doi.org/10.14569/ijacsa.2023.0140449>
- [3] D. G. Kampitaki and A. A. Economides, “Selfishness in Mobile Ad-Hoc Networks: A Literature Review on Detection Techniques and Prevention Mechanisms,” *IEEE Access*, vol. 11, pp. 86895–86909, Jan. 2023, doi: <https://doi.org/10.1109/access.2023.3305262>.
- [4] R. Agrawal *et al.*, “Classification and comparison of ad hoc networks: A review,” *Egyptian Informatics Journal*, vol. 24, no. 1, pp. 1–25, Mar. 2023, doi: <https://doi.org/10.1016/j.eij.2022.10.004>.
- [5] Nursyahira Mohd Salleh and Nor Aida Mahiddin, “MANET Performance Measurement of DYMO Routing Protocol by Varying Density and Mobility Speed,” *International Journal of Engineering Trends and Technology*, pp. 11–18, Oct. 2020, doi: <https://doi.org/10.14445/22315381/cati3p202>.
- [6] A. B. Shaik and S. Srinivasan, “A Brief Survey on Random Forest Ensembles in Classification Model,” *springer*, pp. 253–260, 2019, doi: [https://doi.org/10.1007/978-981-13-2354-6\\_27](https://doi.org/10.1007/978-981-13-2354-6_27).
- [7] J. Ma *et al.*, “De-Anonymizing Social Networks With Random Forest Classifier,” *IEEE*, vol. 6, pp. 10139–10150, Jan. 2018, doi: <https://doi.org/10.1109/access.2017.2756904>.
- [8] P. Casas, “Machine Learning Models for Wireless Network Monitoring and Analysis,” *IEEE Wireless Communications and Networking Conference Workshops*, pp. 242–247, 2018, doi: <https://doi.org/978-1-5386-1154-8>.

- [9] Y. Y. Aung and M. M. Min, "An Analysis of Random Forest Algorithm Based Network Intrusion Detection System ," *IEEE computer society*, pp. 127–135, Jun. 2017, doi: <https://doi.org/978-1-5090-5504-3>.
- [10] B. Bhushan, S. Gupta, and C. K. Nagpal, "Comparison of on Demand Routing Protocols," *International Journal of Information Technology and Computer Science*, vol. 5, no. 3, pp. 61–68, Feb. 2013, doi: <https://doi.org/10.5815/ijitcs.2013.03.08>.
- [11] J. P. Singh and A. Kr. Gupta, "A Review on Dynamic MANET On Demand Routing Protocol in MANETs," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 2, no. 2, pp. 21–26, Apr. 2013.
- [12] V. Y. Kulkarni and D. P. K. Sinha, "Random Forest Classifiers :A Survey and Future Research Directions," *International Journal of Advanced Computing*, vol. 36, no. 1, pp. 1144–1153, Apr. 2013, doi: <https://doi.org/27702358>.
- [13] A. K. Gupta, H. Sadawarti, and A. K. Verma, "Implementation of DYMO Routing Protocol," *International Journal of Information Technology, Modeling and Computing*, vol. 1, no. 2, pp. 49–57, May 2013, doi: <https://doi.org/10.5121/ijitmc.2013.1205>.
- [14] S. K. Bisoyi and S. Sahu, "Performance analysis of Dynamic M formance analysis of Dynamic MANET Ondemand (D T Ondemand (DYMO) Routing protocol ," *International Journal of Computer and Communication Technology*, vol. 2, no. 2, 2011, doi: <https://doi.org/10.47893/IJCCT.2011.1082>.
- [15] S. A. K. Al. Omari and P. Sumari, "AN OVERVIEW OF MOBILE AD-HOC NETWORKS FOR THE EXISTING PROTOCOLS AND APPLICATIONS," *Journal on Applications of Graph Theory in Wireless Ad hoc Networks and Sensor networks*, vol. 2, no. 1, pp. 87–110, Mar. 2010, doi: <https://doi.org/10.5121/jgraphhoc.2010.2107>.
- [16] A. S. Raghuvanshi and S. Tiwari, "DYMO as routing protocol for IEEE-802.15.4 Enabled Wireless Sensor Networks ," *IEEE*, 2010, doi: <https://doi.org/978-1-4244-9730-0/10/>.
- [17] Georgios Koltsidas, Stylianos Karapantazis, G. Theodoridis, and Fotini-Niovi Pavlidou, "A Detailed Study of Dynamic Manet On-demand Multipath Routing for Mobile Ad hoc Networks," *IEEE*, pp. 1–5, Jul. 2007, doi: <https://doi.org/10.1109/wocn.2007.4284181>.
- [18] H. R. Pous and J. H. Joancomart, "Secure Dynamic MANET On-demand (SEDYMO) Routing Protocol," *IEEE computer society*, 2007, doi: <https://doi.org/0-7695-2835-X/07>.