

# **Implementation of DTN Routing in IOT**

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR  
THE AWARD OF DEGREE  
OF  
MASTER OF TECHNOLOGY  
IN  
INFORMATION SYSTEMS

Submitted by:

**SATYAM BAJPAI**

**2K20/ISY/18**

Under the supervision of

**Mrs. Anamika Chauhan**

(Assistant Professor)



**DEPARTMENT OF INFORMATION TECHNOLOGY**

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

MAY, 2022

DELHI TECHNOLOGICAL UNIVERSITY



STUDENT DECLARATION

I hereby declare that the work presented in this report entitled “**Implementation of DTN Routing in IOT**”, in fulfilment of the requirement for the award of the MASTER OF TECHNOLOGY degree in Information Systems submitted in Information Technology Department at DELHI TECHNOLOGICAL UNIVERSITY, New Delhi is an authentic record of my own work carried out during my degree under the guidance of Mrs Anamika Chauhan.

DATE- May, 2022

PLACE- New Delhi

*Satyam Bajpai*

Satyam Bajpai 2K20/ISY/18

DELHI TECHNOLOGICAL UNIVERSITY



CERTIFICATE

This is to certify that Satyam Bajpai (2K20/ISY/18) have completed the project titled “**Implementation of DTN Routing in IOT**” under my supervision in partial fulfilment of the MASTER OF TECHNOLOGY degree in Information Systems at DELHI TECHNOLOGICAL UNIVERSITY.

DATE-May, 2022

PLACE- New Delhi

Guide Signature: Mrs. Anamika Chauhan

## DELHI TECHNOLOGICAL UNIVERSITY



### ACKNOWLEDGEMENT

I am very thankful to Mrs Anamika Chauhan and all the faculty members of the Department of Information Technology of Delhi Technological University. They all provided us with immense support and guidance for the project.

I would also like to express my gratitude to the university for providing us with the laboratories, infrastructure, testing facilities and environment which allowed us to work without any obstructions.

I would also like to appreciate the support provided to us by seniors and our peer group who aided us with all the knowledge they had regarding various topics.

Satyam Bajpai 2k20/ISY/18

## Table of Contents

Declaration.....	ii
Certificate.....	iii
Acknowledgement.....	iv
List of Figures.....	vii
List of Tables.....	viii
Abstract.....	1
Chapter 1: Introduction.....	2
Chapter 2: DTN (Delay Tolerant Network).....	3
2.1: DTN Classification.....	4
2.1.1: Classification of Network Topology.....	4
2.1.2: Classification of Routing Strategy.....	5
2.1.3: Classification of semantic and replication.....	5
2.2: DTN solution's performance evaluation.....	5
2.2.1: Delivery metrics.....	6
2.2.2: Effective metrics.....	6
2.2.3: Resource metrics.....	7
2.3: Literature Review on DTN.....	7
Chapter 3: DTN Routing.....	10
3.1: Epidemic Routing.....	10
3.2: PRoPHET Routing Protocol.....	11
3.3: Spray and Wait Routing.....	11
Chapter 4: DTN with Machine Learning.....	13
Chapter 5: Literature Review of Machine Learning with DTN.....	15
Chapter 6: Algorithm Development.....	18
6.1: Classifiers applied to DTN.....	19
6.1.1: Decision Tree.....	19
6.1.2: XGBoost (Xtreme Gradient Boosting).....	19
6.1.3: AdaBoost (Xtreme Gradient Boosting).....	21
6.1.4: Random Forest.....	22
6.1.4: KNN.....	23

6.1.6: Gaussian Naïve Bayes.....	24
Chapter 7: Implementation.....	25
7.1: Environment Setup.....	25
7.2: Data Preprocessing.....	25
7.2.1: Preprocessing of Node Location data.....	26
7.2.2: Preprocessing of Message Delivery Report data.....	26
7.3: Optimizing the parameters for the Classifiers.....	27
7.3.1: Decision Tree.....	27
7.3.2: KNN.....	27
7.3.3: Gaussian Naïve Bayes.....	27
7.3.4: XGBoost .....	27
7.3.5: AdaBoost .....	28
7.3.6: Random Forest.....	29
7.4: Evaluation Models.....	30
7.4.1: OneVsRest(OvR).....	30
7.4.2: Binary Relevance.....	30
7.4.3: Chain Classifiers.....	30
7.4.4: Ensemble Chain Classifiers.....	31
7.4.5: Label power-set.....	31
7.5: Evaluation Metrics.....	31
7.5.1: Jaccard Similarity Score.....	32
7.5.2: Hamming Loss.....	32
7.5.3: Zero-one Loss.....	33
7.5.4: F-1 score.....	34
Chapter 8: Results.....	36
8.1: Jaccard similarity score.....	36
8.2: Hamming Loss.....	37
8.3: Zero-one Loss.....	38
8.4: F-1 score.....	39
Chapter 9: Conclusion.....	40
References.....	41

## List of Figures

• Fig.1: DTN Classification	4
• Fig 2: Evolution of XGBoost from D-Tree	19
• Fig 3: Evolution of XGBoost from D-Tree	20
• Fig 4: Decision Tree with AdaBoost	21
• Fig 5: Gaussian Normal Distribution	24
• Fig 6: Node location data and derived features from Node location data	26
• Fig 7: Final Output data labels after Preprocessing	27
• Fig 8: Jaccard similarity score for PRoPHET	32
• Fig 9: Jaccard similarity score for Epidemic	32
• Fig 10: Hamming loss for PRoPHET	33
• Fig 11: Hamming loss for Epidemic	33
• Fig 12: Zero-one loss for PRoPHET	34
• Fig 13: Zero-one loss for Epidemic	34
• Fig 14: F-1 score for PRoPHET	35
• Fig 15: F-1 score for Epidemic	35
• Fig 16: Jaccard similarity score graph for PRoPHET	36
• Fig 17: Jaccard similarity score graph for Epidemic	36
• Fig 18: Hamming Loss graph for PRoPHET	37
• Fig 19: Hamming Loss graph for Epidemic	37
• Fig 20: Zero-one loss graph for PRoPHET	38
• Fig 21 Zero-one loss graph for Epidemic	38
• Fig 22: F-1 score graph for PRoPHET	39
• Fig 23: F-1 score graph for Epidemic	39

## List of Tables

Table 1: Classification and discussion of related work.....	9
Table 2: Literature Review of DTN Routing with Machine Learning.....	17



## **ABSTRACT :**

Routing strategies in DTN is basically to choose a suitable node to carry forward the message copies for the successful transfer of message to the destination node. The routing decision, whether to transfer the data to the node encountered or not can be taken as a multilabel classification problem. We have used five multilabel classification techniques for finding the optimum technique for this task on Zebranet UTM-1 data for PRoPHET Routing and Epidemic Routing. The techniques used are Ensemble chain classifiers (ECC), CC(Chain classifiers), BR(Binary relevance), Label Power-set problem transformation and OneVsRest. We have used 7 classifiers as the base learners like XGBoost, AdaBoost, Random Forest, Naive Bayes , Decision Tree, k-NN and MLP. The library used for parameter optimization of the classifiers are hyperopt library and GridSearchCV. Ensemble chain technique with XGBoost classifier on PRoPHET routing data outperformed all the techniques for the PRoPHET and epidemic routing with an accuracy score of 96.1% and Jaccard score of 92.08%.

## CHAPTER -1 INTRODUCTION

There have been massive development in the routing strategies of Delay Tolerant Network over the period of time. The most well known algorithm follow the flooding-based or store-carry forward based strategies for the purpose of routing in DTN. Routing is the effective decision making when a node encounter new node in it's region and have to decide whether to forward a copy of message to that node or not, with the probability that it will reach a destination node. The Authors in the past few years have applied machine learning to determine the best suitable node in a region to share the message copies so that message has the higher probability to reach the destination node. The overall focus is to reduce the overhead and increase the message delivery rate while optimizing the buffer.

In this work we have chosen Naïve Bayes, Decision Tree, k-NN classifiers which have been used earlier in the previous work [27]. We have tried to implement XGBoost, AdaBoost, Random Forest and MLP classifiers in order to compare the results with the already used techniques. XGBoost and AdaBoost are the Gradient Boosting Algorithms while Random-forest is a Bagging and Bootstrapping based algorithm. Bagging is the technique in which multiple D-Trees are produced in parallel and they altogether form the base learners. Boosting have weak learners with high bias as their base learners, whose prediction accuracy is just a little bit high than the random guess. But all weak learners have some important feature that can be helpful in prediction. Boosting technique generates a strong learner by integrating the qualities of weak learners. As in bagging the decision trees are built in a parallel manner, on the other hand boosting form the trees in a sequential manner so that the errors encountered in the previously generated tree are reduced in the subsequent trees.

We have used advance classifiers XGBoost, AdaBoost and Random-Forest in order to compare their performance in the Zebranet UTM 1 dataset for PRoPHET and Epidemic protocol. We have used all the classifiers with the multi-label classification algorithms like Ensemble chain classifiers, chain classifiers, Binary Relevance, Label power-set and OneVsRest. The overview of all the classifiers and multi-label methods have been described in chapter -6.

## Chapter 2: DTN (Delay Tolerant Network)

There have been rapid development in the technologies from the past few years. With these rapid development the new communication systems are required like delay tolerant networks. DTN is a network architecture which deals with the problem of end to end connectivity in a dynamic communication system. DTN deals with solving the problem of lack of continuous network connectivity between the nodes in a network in a dynamic or mobile environment. Over the years many routing protocols have been introduced in support of DTN. So the researchers should try to introduce solutions in order to support emerging network based applications like IOT (Internet of things) applications where DTN techniques can be useful to provide delay tolerant support.

DTN routing protocols are based on Store-Carry-Forward strategy. Suppose a data was lost at a node between the source and its destination during the message transmission, then the message can be stored or saved and then forwarded when the disrupted or disconnected node rejoins the network. The fundamental principle of DTN to forward the data from source to destination is store-carry and forward. Therefore, to route the data from source to destination DTN uses Store-Carry-Forward strategy. In Store-Carry-Forward strategy, an intermediate mobile nodes is required to store message or information that needs to be transmitted until it reestablishes a connection or finds a suitable node in the path to forward the message or information towards the destination [16].

However, maintaining the established connection is not enough but the exchange of data and the protection of data received by the receiving entities is also important. The received data must be understood by the receiving entities. Although many DTN routing protocols have been introduced for DTN networks, but with the advancement in the technologies and introduction of new network based applications like IOT, and drones, the DTN routing protocol has still been useful. So it is required that we should develop a reliable, robust and feasible routing protocol for these new network based applications.

Some of the key features of DTN which varies from traditional networks are as follows:

- Low data rate
- Disconnection
- Limited resources

- Limited longevity
- High latency.

These important features are the reasons for the different and hybrid proposals schemes for DTN architectures and emerging routing protocols. Now we have defined the metrics for the performance evaluation and the criteria of DTN classification.

## 2.1 DTN Classification

Delay Tolerant networks or (DTN) are the mobile or dynamic networks that are unable to have continuous connectivity or end-to-end synchronous path [10]. Different DTN strategies were introduced to deal with the problem of continuous end to end connection. The features of DTN vary from the traditional Ad Hoc networks. Therefore, the researchers introduced new routing protocols can be helpful in providing the needful support to overcome DTN challenges such as long-delay delivery, transient connection and mobility. We have classified the DTN existing schemes into 3 main categories as following:

- Network topology (Deterministic, Stochastic and Coding Based)
- Routing Strategy (Social and Opportunistic)
- Replication and Semantics (Multicast, Unicast and Anycast)

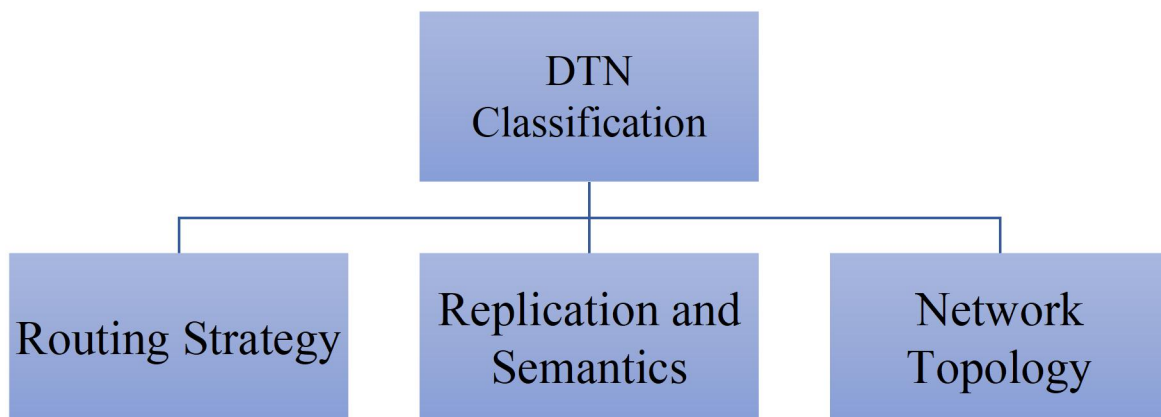


Fig-1: DTN classification

### **2.1.1 Classification of Network topology**

On accessibility of information basis about the topology of network. We classify DTN protocols into coding-based schemes, stochastic and deterministic [8]. In the deterministic category, the schemes made the assumption that info about the network topology is known in advance.

The second category is Stochastic. When the behavior of the communication network is random then the Stochastic Routing protocol is used. This depends on the decisions related to the suitable scenario for forwarding of data. The result differs from sending to any contacts within network to using info like mobility patterns or history data.

The Coding based approach encrypts a multiple original messages or data into one 1 single message with the help of linear combination. This process is called network coding. Another approach id to encrypt one original message into a large number of messages or blocks of codes and this process is called erasure coding. These methods will be helpful in retrieving the original data even when there is loss of subset of code blocks.

### **2.1.2 Classification of Routing strategy**

This class divides the DTN based solutions into opportunistic-based and social-based on the basis of routing strategy. In Social-based protocols, social metrics like, social similarity, betweenness community, centrality, etc. are used by each node for the selection of relay nodes in order to pass the message forward. These metrics describes the proximity and common interest between nodes. On the other hand, in Opportunistic-based routing protocols, selection of relay nodes is done on the basis of parameters like mobility patterns, historic data of the nodes connecting the endpoint, and the chances of connecting the end point.

### **2.1.3 Classification of semantic and replication**

The division of class depends on the message copies amount injected by the starting node in the network either multiple copies or single. Besides this, classification is based on the delivery semantics like, unicast, anycast and multicast. Firstly, it's on the starting node that whether it sends a single and only copy of the data, or it sends multiple copies, therefore the chances of a message reach the endpoint is higher. Depending on the quantity of replicas, DTN routing protocols utilizes network semantic for the delivery of the message. In the Multicast-based

technique we send the data to a collection of needy receivers. On the other hand, unicast-based transmission is based on the delivery of the data to exclusive endpoint. There are basically two methods for choosing the end point node which are “Utility Forwarding” and “Naive Replication”. The Naïve replication method depends on the replication methodology for achieving a successful delivery with the use of multiple message copies. On the other hand, utility forwarding depends on utility metric to succeed encountered node for achieving an efficient and successful forwarding by the use of single copy of message [2]. In anycast technique, the anycast endpoint of the message initialization of data cannot as it may change later and endpoint can be any node within the network.

## **2.2 DTN solution’s performance evaluation**

There are many criterias of evaluating the performance of a DTN solutions. Performance metrics can be classified into 3 basic categories:

- Effectiveness metrics
- Delivery metrics
- Resource metrics

DTN routing algorithms should be developed taking into mind these performance metrics. The developed protocol should increase delivery ratio, decrease the transmission delay and cost, optimize resource allocation and increase data effectiveness. We will discuss about these evaluation factors and we will relate the evaluation factors with IOT features and the issues faced to achieve efficiency.

### **2.2.1 Delivery metrics**

Delivery metrics is an important performance factor in DTN. Hence we describe 3 important delivery parameters and they are delivery cost, delivery latency and delivery ratio. Delivery ratio is the ratio of produced data that efficiently reach their endpoint node. Delivery latency defines latency between the time data was produced and the message reach its final destination. Deliver cost is the measurement of the amount of copies generated for routing of a single message. It is evident that the multiple copies based routing protocol will cost higher than single copies one.

### 2.2.2 Effectiveness metrics

Here the factors affecting the effectiveness of DTN are security, coverage and data. Security effectiveness implies that before a message reaches its unique destination, a message or data can traverse to an arbitrary or a random route. So security is the main concern in DTN and many researchers have worked to analyze access control, anonymity, privacy, etc. Coverage is the percentage of relevant destination nodes that hold a replica of data till TTL (time to live) metric of the concerned data is expired. Data effectiveness is the ratio of data traffic produced when multicast messages efficiently reach the destination node.

### 2.2.3 Resource metrics

An important issue related to smart objects in general IoT applications and mobile devices in DTN is the limitation of resources. These resources are generally memory, bandwidth and battery. Hence, we now provide two main parameters of resource allocation efficiency.

**Energy:** This is the measurement of the energy dissipated in order to deliver the message successfully. The energy consumed is totally depended upon the amount of copies generated and the amount of messages that are delivered to the final endpoint node.

**Overhead:** Resources metrics like memory storage and Bandwidth are also the main factors or parameters for the evaluation of DTN strategies. Overhead is the measurement of the ratio between the total number of message transmission needed for the delivery to the messages delivered.

## 2.3 Literature Review on DTN

The following table shows the survey of the researchers. Some researchers focused on the Design issues while other focused on the taxonomy of various routing protocols. Zhang [8] does the categorization a significant number of unicast routing algorithm introduced till 2006 on the basis of their mobile model. Deterministic strategy permits to send data forward having knowledge of upcoming topology of network. The routing schemes that comes under this class are modified shortest path, space time, and tree-based. On the other hand, stochastic strategies does not have knowledge of upcoming network status. Therefore, no prior forwarding can be done. Following are some of stochastic strategies: epidemic and randomized flooding, coding-

based, controlled- movement-based, model-based, and history-based. Likewise, Shen et al. [6] classified Delay tolerant network routing protocols in DTN into forwarding type and flooding type techniques simultaneously. In flooding strategy every node has a multiple instance of copies and sends them to a group of nodes for a single message. Whereas in Forwarding technique needs apriori info of the network for selection of optimized paths In addition to flooding and forwarding routing strategies D’zousa & Jose [3] published another strategy based on using special device in their paper. In this Approach, additional device either stationary or mobile are used to effect communication network. JS and Abraham [1] classified routing strategies into 2 main categories: forwarding and replication. Khabbaz et al. [4] gave a broad view about DTN design challenges. Formerly, authors classified forwarding strategies introduced from 2007 to 2010 into resource allocation-based, encounter-based, probabilistic, opportunistic, network coding-based, delegation and spray-and-wait and vector-based, load-balancing based.

<b>Year</b>	<b>Ref</b>	<b>Paper Overview</b>
2006	[8]	<ul style="list-style-type: none"> <li>•Mobility and topology-based classification ( deterministic, stochastic &amp; coding-based classification)</li> <li>•Survey of unicast DTN schemes up to 2006</li> </ul>
2008	[6]	<ul style="list-style-type: none"> <li>•No taxonomy</li> <li>•Classification in Flooding and Forwarding</li> </ul>
2010	[3]	<ul style="list-style-type: none"> <li>• special device-based, history-based, &amp; flooding-based Classification</li> </ul>
2010	[5]	<ul style="list-style-type: none"> <li>•Classification on the basis of Architectural layer</li> <li>•Taxonomy of DTN till 2010</li> <li>•DTN design problems</li> </ul>
2012	[1]	<ul style="list-style-type: none"> <li>•Classification on the basis of Routing strategies</li> <li>•DTN issues analysis</li> </ul>
2012	[4]	<ul style="list-style-type: none"> <li>•Cooperative DTN routing protocols</li> <li>•Taxonomy of DTN between 2007 and 2010</li> </ul>
2012	[2]	<ul style="list-style-type: none"> <li>•Taxonomy between 2006 and 2010</li> <li>•Utility forwarding and Naïve Replication Classification</li> <li>•Classification on Delivery semantic</li> </ul>



2013	[9]	<ul style="list-style-type: none"> <li>•Classification of social based routing</li> <li>•Positive and negative social properties</li> <li>•Taxonomy between 2007 and 2011</li> </ul>
2016	[7]	<ul style="list-style-type: none"> <li>•Classify on the basis of network primitives and message replication</li> <li>•Protocols- Pure Opportunistic, dissemination and social based</li> </ul>
2018	[33]	•“Fcms: a fuzzy routing-forwarding algorithm exploiting comprehensive node similarity in opportunistic social networks”.
2019	[29]	<ul style="list-style-type: none"> <li>•Discussion on Spray-and-wait and finding new Metric for finding the suitable node based on QON(Quality of node)</li> <li>•Adaptive Spray-and-wait or QON-ASW.</li> </ul>
2019	[30]	•Delivery probability-based ASW.
2019	[31]	•“A motion awareness based routing algorithm for delay tolerant network”.
2021	[32]	•“An adaptive multiple spray-and-wait routing algorithm based on social circles in delay tolerant networks”.

Table 1: Classification and discussion of related work

## CHAPTER -3 DTN ROUTING

Routing signifies different routes that data bundles take on their way to a target destination. In other words we can say data travelling on the internet over delay tolerant network used by mobile or other digital communications. Researchers developed several routing protocols and algorithms. Some of them are listed below.

### 3.1 EPIDEMIC ROUTING

This algorithm is based on the concept of replication. It comes under the category of flooding based scheme.

- In this algorithm each node transmits and share the copy of its message to newly discovered node.
- Again this newly discovered node do the same thing with its messages .
- There are different types of epidemic routing protocol like Epidemic with immunity table, Epidemic with encounter, Epidemic with TTL, P-Q epidemic.

Advantages:

- Little to no reliance on special nodes.
- Low delays
- Simplicity

Challenges:

- This algorithm requires boundless buffer size.
- Limitless energy to give high deliverance rate.
- All those intents and purposes this conditions are difficult to be applied .

## 3.2 P<sub>RO</sub>PHET ROUTING PROTOCOL

In Prophet routing protocol [22] delivery predictability is defined as estimate probabilistic metric i.e.

- $P(\text{node\_A}, \text{node\_B})$ , at each node a for each destination node b. whenever two nodes meet in the network scenario it swaps the summary vector which consist of delivery predictability values.
- After swap process, all nodes updates their own delivery predictability in the summary vector.
- A low predictability value is assigned to the node if the contacts between two nodes are very rare or no contact exists between two nodes.
- If the nodes are meet regular interval then it delivery predictability is very high.
- The transitivity property of Prophet routing protocol state that if node\_A regularly meets B and node\_B regularly meet node\_C, then C is appropriate node for A, hence A marks C delivery predictability value as high in the summary vector.

### Challenges:

- The messages reaching the destination node might be corrupted due to the malicious path nodes.
- The likelihood of the messages to be aborted is high as there are too many nodes in the path of source to destination.
- There might be a situation that few messages are dropped due to TTL expiration and congestion of buffer.

## 3.3 SPRAY AND WAIT ROUTING

SAW is the mixture of replication based routing and direct transmission and forwarding based routing protocols. The high deliverance rate of replication based(Epidemic) and resource utilization feature of forwarding based technique helps SAW to perform better. SAW works in two phases.

- a- First, at the spray phase: let the source node have message M, then for each message M, N number of copies are transferred to N adjacent nodes or relay nodes.

b- Second phase is the wait phase in which each of the source node will keep a copy of M until the target node is achieved or TTL is expired. The N constant is to be defined at the beginning of the simulation.

**Advantages:** SAW has the benefit or both replication-based and flooding based routing techniques.

**Challenges:**

- The nodes with the high metric value, might not be able to transfer the message forward to the other relay nodes due to it's poor ability which in turn result in message dropping and hence reducing the message delivery rate.
- Suppose the connection between the nodes is for a very short duration, so considering only the high metric value and not considering the connection time between the nodes may result in the unsuccessful transfer of messages from one node to another. The connection time should be long enough such that the messages are transferred successfully between the nodes. If connection time is not enough then even the selection of nodes on high metric value can face the message delivery rate reduction.

## CHAPTER 4: DTN with Machine Learning

The focus of this work is to optimize the routing task in DTN with the help of machine learning to decrease the overhead and choose a best node in the network to forward a message. This implementation of machine learning in DTN Routing will help us solve the various scenarios of a DTN like entering of the new nodes in DTN, excluding or removal of a node from the network and for nodes involved or working in different time[41]. The use of machine learning will help us determine or predict the pattern or behavior of delay tolerant network and help us choose the node to transfer message accordingly.

Machine learning have various methods which can be used for the routing purpose of DTN. Machine Learning is a branch of artificial intelligence which has the capability of self-learning and improving the performance of the model with the help of without hard coding the results. The model learns with the help samples i.e. data or observations to find the patterns in our data which are basically helpful for defining the decision boundaries. The machine learning algorithms which are commonly used are supervised, unsupervised and reinforcement learning.

**Supervised machine learning** approach is applied for the labelled dataset. This approach is applied on the observations or data which are labelled or data classified in classes. The model is trained based on the observations of this data and exploring a pattern from this data and the trained model is used to predict the test data which is usually the data which is not trained on the model. The model is used to forecast the future predictions. The inputted dataset with labels given to the model is called as training data, which in turns create an inferred function which is finally used to predict the outcomes of the nontrained or new unseen data. The predicted output can then be compared to the expected output for the unseen data and different evaluation metrics can be calculated with the help of that and modify the model with the help of back-propagation. Random forest is an example of supervised learning.

In **unsupervised learning** the data in dataset need not be labelled or categorized in classes. Unsupervised learning finds or explore a hidden pattern from the dataset instead of predicting the output. Clustering and association are the further divisions of unsupervised learnings. K-means clustering is an example of unsupervised learning.

**Reinforcement learning** consists of the models which make use of the estimated errors which are passed as the penalties or rewards to the model to train them. The quantity of error determines whether the penalty is high or low or the reward is low or high. If the estimated error is low, then reward is high and penalty is low and if the estimated error is high, then the reward is low and penalty is high. The models in reinforcement learning requires reward feedback as reinforcement signal which is helpful in determining which action is best .

The performance of the model can be calculated with the help of different evaluation metrics like F1-score, confusion matrix, AUC-ROC, Root Mean Squared error, Log Loss, etc. The evaluation metrics used in this work will be discussed in the further chapters.

## **Chapter 5: Literature Review of Machine Learning with DTN Routing:**

The work of applying machine learning techniques to solve the DTN routing issues have been addressed by many authors [15], [16], [17], [18], [19]. A lot of experimentation has been done in this approach. The routing in the DTN can make use of the mobility patterns in the DTN as most real life DTNs follow kind of periodic repetition or pattern for example movement of a vehicle from one point to another on the daily basis. The temporal and spatial data while transferring the messages from one node to another was proposed in [15]. In [15] the author used DTN routing framework which was based on Bayesian classifier for routing. The mobility traces from a public transport bus network were used for the simulation of a real world vehicular DTN network, which infers that the basic single copy forwarding technique which utilizes this framework can perform much better than the gradient based single copy technique by the margin of 25% in context to delivery ratio.

In [16] the author implemented the classification mechanism on the two commonly used DTN routing epidemic and spray-and-wait for the performance evaluation on Seattle buses. The classifier used was Decision Tree classifier. The performance in SAW after applying the classification mechanism improved due to the fact that the fewer number of copies were forwarded by identifying the more suitable nodes. In case of the epidemic routing after applying the classification mechanism the overhead was reduced without having an adverse effect the chances of message delivery. The use of limited copies had resulted in the increase of delivery delay. Multi-copy routing scheme was also used which demonstrated that the real world traces faced the greater overhead as compared to the simulated environment because of the dynamic distribution of nodes.

[17] discussed the machine learning concept and techniques to solve the routing issue in delay tolerant space networks. The routing algorithm used is Contact Graph Routing and the machine learning concepts applied are Bayesian learning and Reinforcement learning. The author described CGR, Naïve Bayes classification, and CGR in this paper.

In [18] the algorithm based on Q-Routing was used for packet routing. In Q-routing, the machine learning based technique- reinforcement learning was used along with every node of the network. The nodes used the local communication to collect the correct information on the basis of which

the correct routing decisions are made which result in decrease in delivery time. The 36-node, dynamically connected network with varying network load experiment demonstrate that Q-Routing algorithm outperforms the non-adaptive algorithms like precomputing the shortest path. Q-Routing algorithm doesn't require the traffic pattern or topology of the network in advance.

The performance for a particular network traffic configuration can degrade or fail if the traffic changes. The solution of this issue can be the optimization of the routing scheme on the basis of the past behavior of the network hoping that the improved routing configurations will work fine in the future dynamics of the network. Other solution is that the configurations of static routing scheme are optimized on the large dynamic scenarios of the network. Machine learning techniques can be another solution, which can make use of the pattern of the past network patterns to train the model on it and improve the routing decisions in order to reduce the delay and overhead in the future network pattern. [19] makes use of Softmin routing scheme. In this approach, the algorithm makes use of the previous network matrices to find the edge weights.

[27] presents the DTN implementation with machine learning which used multilabel classification scheme such as Ensemble of Chain Classifiers, Label Powerset, Chain Classifiers and One-versus-All. The classification algorithm used are KNN, Naïve Bayes and D-TREE. The routing protocols used are epidemic and prophet. Evaluation metrics used for the performance evaluation are jaccard similarity score, hamming loss, zero-one loss and F-1 score.

[26] presents the epidemic and prophet routing algorithms in DTN which uses machine learning in order to predict the neighbour nodes. These neighbour nodes are predicted on the basis of delivery status of the history messages using machine learning techniques. IBR-DTN Bundle protocol implementation is used to get the real world data of the network. The emulator used was CORE(Common Open Research Emulator). The classification algorithms used are K-Nearest Neighbors, Naïve Bayes, Decision Tree, and multi-label methods such as Ensemble of chain classifiers, label powerset, etc.

[28] presents the improved routing algorithms for interplanetary DTN. The most popular routing algorithms for opportunistic and deterministic DTN are PROPHET ( Probabilistic Routing Protocol using History of Encounters and Transitivity ), SAW (Spray-and-Wait), RAPID (Resource Allocation Protocol for International DTN) and DTLSR (Delay Tolerant Link State Routing ). CGR (Contact Graph Routing) is a well known algorithms in the field of



interplanetary networks. CGR utilizes the contact time and distance information of the network. This paper works on the router based on machine learning techniques like Reinforcement learning and Bayesian learning in order to make the improved routing decisions of CGR. This paper also discussed about CGR, Naïve Bayes classification.

<b>References</b>	<b>Machine Learning Method</b>	<b>Routing Protocol</b>
[15]	Bayesian Classifier	Gradient based single copy
[16]	Decision Tree	Epidemic, Spray-and-wait
[17]	Naïve Bayes Classifier, Reinforcement	Contact Graph Routing
[18]	Reinforcement Learning	Q-Routing
[19]	Reinforcement learning	Softmin Routing
[27]	KNN, Naïve Bayes, Decision Tree, multilabel classification methods	Bundle protocols, Epidemic, PRoPHET
[26]	KNN, Naïve Bayes, Decision Tree, multilabel classification methods	PRoPHET, Epidemic
[28]	Reinforcement learning, Bayesian Learning, Naïve Bayes Classification	Contact Graph Routing

Table 2: Literature Review of DTN Routing with Machine Learning

## Chapter -6 Algorithm Development:

In this work we have proposed the usage of Bagging and gradient boosting techniques for improving the routing decisions in DTN routing protocols. The focus of this work is to optimize the routing task in DTN with the help of machine learning to decrease the overhead and choose a best node in the neighboring network of a node to forward the message copies. Machine learning has various methods which can be used for the routing purpose of DTN. Machine Learning is a branch of artificial intelligence which has the capability of self-learning and improving the performance of the model with the help of without hard coding the results. The model learns with the help of samples i.e. data or observations to find the patterns in our data which are basically helpful for defining the decision boundaries. This implementation of machine learning in DTN Routing will help us solve the various scenarios of a DTN like entering of the new nodes in DTN, excluding or removal of a node from the network and for nodes involved or working in different time. The use of machine learning will help us determine or predict the pattern or behavior of delay tolerant network and help us choose the node which is suitable to carry the message forward accordingly.

In accordance with the trained model, we classify the nodes in the path that constitutes the best possible path to the target node. We make the assumption that the behavior of each node conforms to a periodic pattern that can be predicted which depends on various characteristics like the set of surrounding nodes, length of the connection time between two nodes, the buffer capacity, the data rate and so on. This time span is known as an epoch, and it is divided up into several time slices. We are determining, with regard to a message, if the message has been sent to the  $i^{th}$  node, where  $i$  falls within the range  $[0, \text{numNodes}]$ . Therefore, what we have here is a single multi-label classification issue that has been broken down into many binary classification problems. In addition, it assigns classes to each label separately, excluding the interdependency factor of the labels of the output. The performance problem of multilabel classification may be solved using machine learning using Ensemble Classifier Chains (ECC), which also takes interdependence into consideration

The machine learning algorithms which are mainly used for the purpose of classification task are described below.

## 6.1 Classifiers applied to DTN:

### 6.1.1 DECISION TREE:

Decision Tree or D-Tree is a supervised learning algorithm which can perform the regression and classification task. D-Tree based models are trained on the data features and predict the output on the basis of the simple decision rules. D-trees are simple and flexible. Simple because the complexity of using the D-Tree is logarithmic. D-Trees are capable of handling the categorical as well as numerical data. D-Trees might have the problem of high variance which can lead to overfitting of the model.

### 6.1.2 XGBoost (Extreme Gradient Boosting):

It is an ensemble ML technique based on D-Tree which utilizes the gradient boosting framework. XGBoost is a supervised learning approach which can be used for classification and regression task. Structured or table data upto medium size, shows the best result on applying the D-Tree based algorithms while for medium to large datasets, ANN shows the best result in terms of prediction problems. XGBoost algorithm has evolved from the Decision Trees over the period of time.

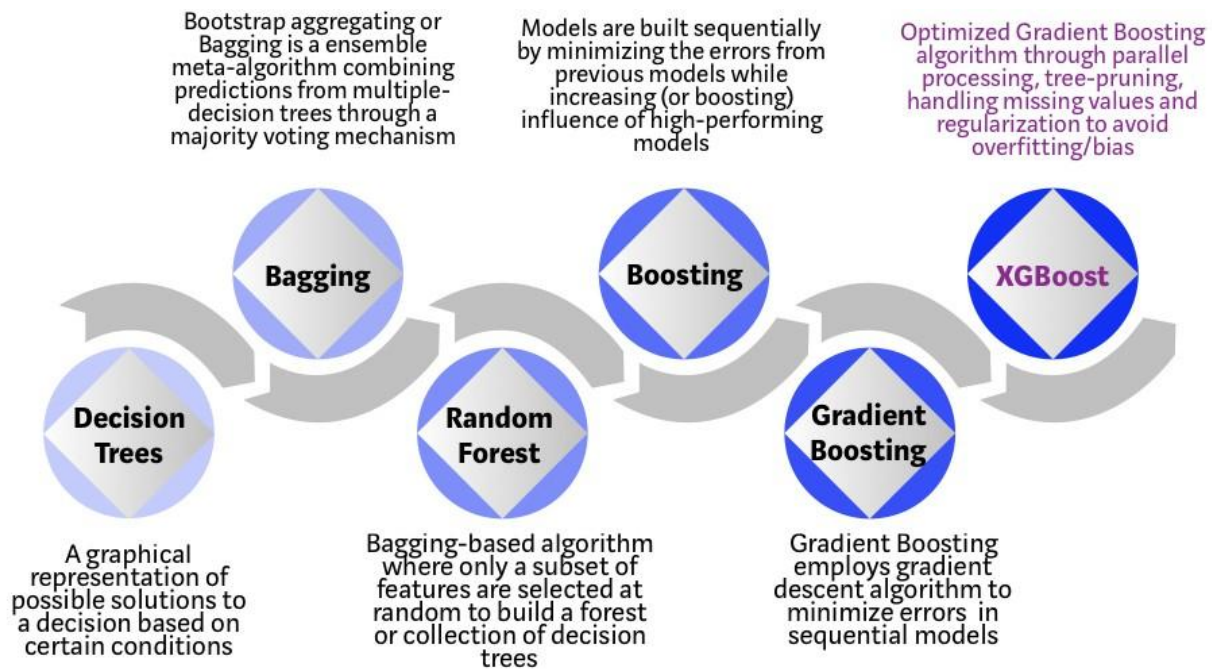


Fig 2: Evolution of XGBoost from D-Tree [35]

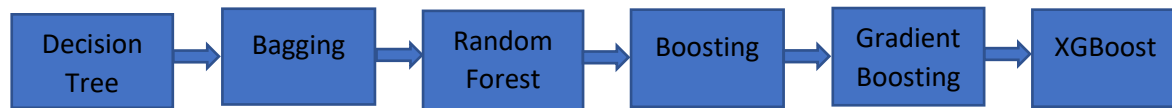


Fig 3: Evolution of XGBoost from D-Tree

The basic architecture of Gradient Boosting Algorithms differ from other boosting algorithms in the way that here loss function optimization is done instead of increasing misqualified branches weights. XGBoost provides the support for cross validation, cache optimization, parallel processing, tree pruning and can take care of outliers and missing values. XGBoost has the feature managing the memory efficiently for big datasets to tackle RAM exceeding problem.

### Features of XGBoost:

XGBoost is an ensemble machine learning technique which provides an efficient solution to use the prediction qualities of multiple machine learning models instead of a single machine learning model because there are several cases where using only one machine learning model may not give satisfactory results. The ensemble of the models are basically called the base learners which can be from one learning algorithms or more than one learning algorithms. The two most popular ensemble learners are Bagging and Boosting which are most widely used with D-Trees. Other than D-trees other statistic models can also be used . Bagging and Boosting are helpful in reducing the variance in the base learners. The result of ensemble techniques is a single model which have the collected power of all models.

**Bagging** is the technique in which multiple D-Trees are produced in parallel and they altogether form the base learners. Suppose there is a dataset and we divide this into two randomly. Now, these randomly split datasets are used to produce two models by training on the D-Tree. D-Trees have high variance due to the fact that when we try to fit these models, the results would be different. Bagging is helpful in decreasing this variance because the sample data is given to the base learners with sample and the final result will be the average of result of all learners.

**Boosting:** Boosting have weak learners with high bias as their base learners, whose prediction accuracy is just a little bit high than the random guess. But all weak learners have some important feature that can be helpful in prediction. Boosting technique generates a strong learner

by integrating the qualities of weak learners. Now the final strong learner is able to reduce the bias as well as variance.

As in bagging the decision trees are built in a parallel manner, on the other hand boosting form the trees in a sequential manner so that the errors encountered in the previously generated tree are reduced in the subsequent trees. Every subsequent tree learns from the previously generated tree and update the residual error and the next tree in the sequence learns from the updated residual errors.

Boosting make use of less number of splits as compared to bagging where trees are expanded to the maximum level because the trees with lesser splits are highly interpretable and the trees with higher splits can cause the problem of overfitting. The optimization can be done like determining the tree depth, the learning rate of the gradient boosting, `n_estimators`, `min_child_weight`, `reg_lambda`, `subsample`, `colsample_bytree`, `gamma`, etc.

### 6.1.3 AdaBoost:

AdaBoost is another ensemble based Boosting algorithm in which make use of the boosting technique to integrate the powers of the weak learners and as a resultant generate a strong learner. In AdaBoost algorithm, the D-Tree is known as Decision stumps because the decision tree have only one split.

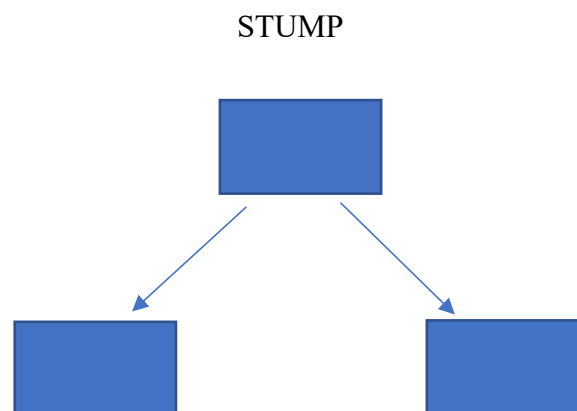


Fig 4: Decision tree with AdaBoost

The procedure of boosting is that, first a model is trained on dataset, then a second model is trained to reduce the bias, variance, errors in the first model and the process continues until the prediction results are correct or the residual errors are minimum. AdaBoost is commonly associated with the Decision Tree. Suppose we have a dataset and on that dataset we have applied kNN, Decision trees and linear regression. The accuracy score of all the 3 models came out to be 70%, 85% and 65%. Here we see that the accuracy score of the models is different on same dataset. This is where Boosting is applied and power of all three models is integrated together to predict the results and the accuracy score will be average of the results of these models.

The AdaBoost algorithm firstly allocate the same weights to all the data points and the model is trained and the predicted output is generated. Now the datapoints which are wrongly predicted are given the higher weights in comparison to the correctly predicted data points. These data points with higher weight have priority in the subsequent model. This process keeps on repeating unless the errors are minimized or the classification is done correctly.

#### **6.1.4 Random Forest:**

As discussed above in Decision Trees article, the D-Tree looks simple and flexible but it may have the high variance problem due to its greedy nature of algorithm which makes our model overfit. Variance is calculated in a different manner for both the classification problems and the regression problems.

In regression problems, the actual variance is calculated which is the variance of the true output from the predicted output and if the expected outputs of the tree is too far from the predicted output then the tree is said to be overfitted as it has high variance. While in classification problems, high generalization error is calculated and if this error is high then we can say that the model is overfitted as it doesn't performs well on the test dataset.

Bagging is the technique which is used to reduce the variance problem and thus preventing the overfitting of model which is described above. Bootstrapping is a part of the bagging technique which is used to make the multiple copies of data with replacement. We can also call it as resampling method of data.

Random Forest is used to solve the issue of high variance as in this we are training a forest of Bagged D-Trees instead of single D-Tree. Random Forest has all the benefits of the D-Trees and is also capable of handling different type of datasets without preprocessing. Random Forest chooses a subset of features randomly instead of all features for the splitting task. Therefore the Tree will be different as algorithm will evaluate less number of outputs.

### 6.1.5 KNN:

KNN stands for K nearest neighbors which is used to classify a new case on the basis of the similarity measure. Similarity measure is the distance functions which can be Manhattan distance, Euclidean distance and Minkowski distance. The new case is classified to the class to which most of it's K nearest neighbors belongs to. The K nearest neighbors are calculated using the distance function. When the value of K is 1, that means the new case is classified to nearest neighbor's class.

Euclidean, Manhattan and Minkowski distance functions are only applicable for continuous variables. Hamming distance can be used for the categorial variables. There can be datasets which have the mix of categorial and numerical values. This issue is solved by the standardization of numerical values in the range of 0 to 1.

$$\text{Euclidean distance} = \sqrt{\sum_{i=1}^k (x_i - y_i)^2} \quad (1)$$

$$\text{Mahattan distance} = \sum_{i=1}^k |x_i - y_i| \quad (2)$$

$$\text{Minkowski distance} = \left( \sum_{i=1}^k (|x_i - y_i|^q) \right)^{\frac{1}{q}} \quad (3)$$

In General the K value between 3 and 10 have been optimal for the majority datasets. But the K value can be optimized by the analyzing the dataset or by the method of cross-validation.

### 6.1.6 GAUSSIAN Naïve Bayes:

Naïve Bayes is a bayes theorem based supervised ML algorithm. Naïve base can be used for the complex problems of classification when the input dimensionality is large. Bayes theorem is the calculation of the conditional probability. Gaussian Naïve Bayes is the Naïve Bayes classifier which is used for continuous data and the data distribution is Gaussian normal distribution.

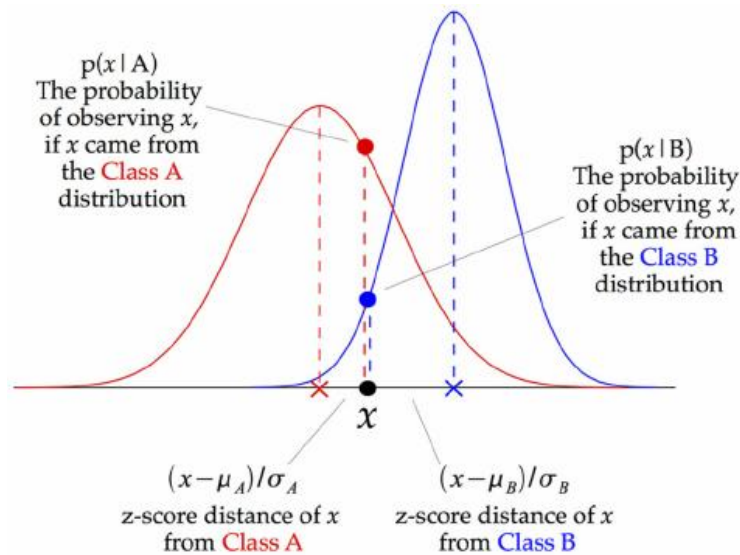


Fig 5: Gaussian Normal Distribution[36]



## **Chapter -7 Implementation:**

We have chosen basically seven classifiers to be part of our implementation for the task of classification namely k-NN, D-Tree, Naïve Bayes, XGBoost, ADABOOST, Random Forest Random Forest is chosen as we want to include one Bagging and Bootstrapping technique where as XGBoost and AdaBoost are taken in order to cover the Gradient Descent Boosting techniques. Since we have Labelled data so we can use supervised learning techniques. The models are build using the multilabel classification techniques namely: ONEVsREST, Binary Relevance, Label Powerset, Chain and Ensemble Chain classifiers. The model is trained such that it finds out the suitable node to transmit the message copies to enhance the capability of the node in finding the best nodes in the region to transmit the message copies to reach the destination node.

### **7.1 Environment setup:**

- ONE Simulator(The Opportunistic Network Environment) for the simulation task.
- DTN2
- Zebranet UTM-1 mobility traces
- The reports generated from the simulation are stored in excel which are used as data.
- In this implementation we have used PROPHET message delivery report, Epidemic Message delivery report, PROPHET Node location report and Epidemic Node Location Report.

We have to write a script file and change the configuration file in order to get the message delivery report and node location report or we can directly apply ML techniques on the already generated report.

### **7.2 Data Preprocessing:**

The Node location dataset and the message delivery report dataset generated with the Zebranet UTM1 mobility traces is preprocessed to get the features and data in the required format so that the models can be build using that.

### 7.2.1 Node location data preprocessing:

The node location dataset for PProPHET and Epidemic is as below. The features in the Node Location Dataset are time, node x and y. the primary goal is to derive the region code for every node. K-Elbow visualizer is used to find the optimum value of k. K-Elbow visualizer is run from 3 to number of nodes in the simulation. The optimum value of k comes out to be 5 which is used as the number of clusters.

Node location dataset has following features (time, node, x, y) from which cluster\_labels are found and these are renamed as region code so the final output label is (node, region)

time	node	x	y	node	region
0	25	p0	359.2952 593.5965	0	3
1	25	p1	595.1697 864.0757	1	0
2	25	p2	150.4684 323.1847	2	1
3	25	p3	246.6003 679.0803	3	6
4	25	p4	99.0202 383.6092	4	1




Fig 6: Node location data and derived features from Node location data

### 7.2.2 Message Delivery Report data preprocessing:

(time, ID, size, hopcount, delivery Time, fromHost, toHost, remainingTtl, isResponse, path)

The preprocessing of the message delivery report involves the unpacking of tuple, assigning all data entries with a time-slice-index by choosing a random epoch size ( we have chosen 400), dropping some of the columns like hopcount, deliveryTime, remainingTtl, isResponse, size and unpack the “fromHost”, “toHost”, “path” and “ID” to the accepted format. Now these columns are formatted in the suitable format. Now all the entries of the messages are grouped in an epoch to produce the node that visit in the time of that epoch. “deliveryStatus” column is created to point out whether the message reached the destination node or not. Then the datasets are merged together to associate the “deliveryStatus” and “regioncode” with the source node and destination node. Now the final output label columns are produced.

	ID	epoch_index	fromHost	toHost	path	node	region
0	1	2	6	1	{1, 7, 5, 6, 0}	6	3
1	1	5	6	1	{3, 6}	6	3
2	1	11	6	1	{8, 3, 6}	6	3
3	1	18	6	1	{8, 4, 3, 6}	6	3
4	1	24	6	1	{2, 4, 6, 8, 3}	6	3



	epoch_index	fromHost	toHost	region_x	region_y	deliveryStatus	d0	d1	d2	d3	d4	d5	d6	d7	d8	d9
0	0	6	1	3	1	False	True	False	False	False	False	False	True	False	False	False
1	3	6	1	3	1	False	False	False	False	True	False	False	True	False	False	False
2	4	6	1	3	1	False	True	False	False	False	False	True	True	True	False	False
3	5	6	1	3	1	False	True	True	False	False	False	True	True	True	False	False
4	8	6	1	3	1	False	False	False	True	True	True	False	True	False	False	False

Fig 7: Final Output data labels after Preprocessing

After preprocessing the dataset is split into the testing and training datasets. The optimum test size can be chosen with the experimentation. We have chosen the test\_size to be 0.15.

### 7.3 Optimizing the parameters for the classifiers:

**7.3.1 Decision tree:** No parameter optimization

**7.3.2 KNN:** Can check with different values of k, the optimized value came out to be 1.

**7.3.3 Gaussian Naïve Bayes:** No parameter optimization is done

**7.3.4 XGBoost:** Parameter Optimization is done using the hyperopt library.

The best XGBoost hyperparameters for PROPHET are :

- colsample\_bytree: 0.833674390498152
- gamma: 0.15237568804619786

- learning\_rate: 0.610283668841475
- max\_depth: 70.0
- min\_child\_weight: 0.0
- reg\_lambda: 0.25588301860187623
- subsample: 0.6016287987826826

The best XGBoost hyperparameters for Epidemic are :

- colsample\_bytree: 0.8422726512110625
- gamma: 0.9486058704052276
- learning\_rate: 0.12804497937652803
- max\_depth: 46.0
- min\_child\_weight: 0.0
- reg\_lambda: 0.4218240114269855
- subsample: 0.7638303806137278

**7.3.5 AdaBoost:** Parameter Optimization is done using the hyperopt library.

The best AdaBoost hyperparameters for PProPHET are :

- learning\_rate: 0.1
- n\_estimators: 198

The best AdaBoost hyperparameters for Epidemic are :

- learning\_rate: 0.1
- n\_estimators: 198

**7.3.6 Random Forest:** Parameter Optimization is done using RandomizedSearchCV.

The best Random Forest hyperparameters for PProPHET are :

- bootstrap=True
- max\_depth=100, max\_features=auto

- min\_samples\_leaf=1
- min\_samples\_split=2
- n\_estimators=800

The best Random Forest hyperparameters for Epidemic are :

- bootstrap: True
- max\_depth: 100
- max\_features: 'auto'
- min\_samples\_leaf: 1
- min\_samples\_split: 2
- n\_estimators: 800

**7.3.7 MLP Classifier:** Parameter Optimization is done using GridSearchCV. MLP classifier is used believing that it will perform better but the results are not as expected.

Best parameters found for MLP classifier in case of PRoPHET are:

- activation: relu
- alpha: 0.05,
- hidden\_layer\_sizes: (50, 100, 50)
- learning\_rate: 'constant'
- solver: 'adam'

Best parameters found for MLP classifier in case of Epidemic are:

- activation: relu
- alpha: 0.05
- hidden\_layer\_sizes: (50, 100, 50)
- learning\_rate: 'adaptive'
- solver: 'adam'

## **7.4 Evaluation Models:**

DTN routing with machine learning can be considered as multilabel classification problem which can be divided into many problems involving binary classifications. The multilabel classification methods used are:

### **7.4.1 OvR (OneVsRest):**

OneVsRest or OvR is a multi-class strategy and can be used for the multilabel classifications as well. OVR can be imported from the `sklearn.multiclass`. OVR is also referred to as One-vs-all as this technique fits 1 classifier per class. Since only one classifier is the representative of a particular class so it makes OvR highly interpretable, and the class information can be extracted properly.

### **7.4.2 Binary Relevance:**

The technique of dividing the classification of routing decisions problem into many problems such that one classifier generates the result as whether a relay node is part of the cluster of nodes in a path or not is said to be Binary Relevance method [14]. Binary Relevance is a method which makes use of problem-transformation method to solve multilabel classification problem. Here the one classification task having multiple output results has been split into multiple classification task.

Multilabel classification algorithms are developed taking into account the interdependency of the binary outputs. Binary Relevance is somewhat same as OvR as in this classifies the labels in an independent manner and without considering the interdependency of the binary outputs.

### **7.4.3 Classifier Chains:**

Classifier chains[13] (CC) is another technique which is used for the multilabel classification task. In this chains of binary classifier is used and the predicted output of the previous classifiers is taken into consideration for the subsequent classifier prediction. The classifiers count is considered to be same as the number of classes. The binary label relevance of the prior steps would have an effect on the subsequent classifiers result. This may give different accuracy in different ordering of label. So the label order sequence can be an issue in the Classifier Chains model.

#### **7.4.4 Ensemble chain Classifiers:**

The ECC is same as the chain classifiers while removing the dependency on the label order of the chains by randomization of the label order and creating multiple chains[27]. This solves the propagation of the errors from the previous steps in the upcoming steps[13]. The binary classifiers alone are not called as the multilabel classifiers, that's why ensemble refers to as the ensemble of multi-label classifiers as it uses multiple classifiers in random label order[12].

#### **7.4.5 Label Powerset:**

Other than Binary Relevance, Label power-set method is also problem transformation method [20, 21,12]. Label Powerset is also used for the task of multilabel classification. The methodology behind the Label Powerset is to merge multiple labels into one single label and turn the problem to the single-labelled-classification task. The single-label or atomic-label subsets is formed from the multi-label problem's labels and this subset contains all the unique subset of labels. In this way Label power-set takes into consideration the interdependency of all the labels. The possibility of the subset of labels may be high, so this might create an issue. So [22] introduced pruned problem transformation (PPT) and [23] introduced HOMER to deal with this issue.

### **7.5 Evaluation metrics :**

Evaluation metrics for the validation of the mutli-label classification are Jaccard Similarity Score[24], Hamming Loss[24], F-1 score[25] and Zero-one loss[24]. These are the metrics we have used for validation of our models OvR, Binary relevance, Label power-set, Chain classifiers and Ensemble Chain Classifiers.

#### **7.5.1 Jaccard similarity score:**

Jaccard similarity score is defined as the 2 label-sets intersection size divided by the two label-sets union size. Score 1 in jaccard is considered to be optimum score and score zero(0) to be worst. The jaccard similarity score of the PRoPHET and EPIDEMIC for various models is shown in below table.

Jaccard	XGBoost	AdaBoost	D-Tree	Naive Bayes	KNN	Random Forest	MLP
<b>OVR_results</b>	0.916593	0.612964	0.890938	0.383533	0.849564	0.872294	0.667028
<b>chain_av_results</b>	0.91266	0.612634	0.89593	0.433006	0.849564	0.874057	0.676387
<b>ensemble_results</b>	0.919344	0.611734	0.905975	0.482701	0.849564	0.876119	0.694087
<b>powerset_results</b>	0.876766	0.471317	0.902273	0.563067	0.849564	0.858604	0.659168
<b>BinaryRelevance_results</b>	0.916593	0.612964	0.889223	0.383533	0.849564	0.87361	0.666963

Fig 8: Jaccard similarity score for PProPHET

Jaccard	XGBoost	AdaBoost	D-Tree	Naive Bayes	KNN	Random Forest	MLP
<b>OVR_results</b>	0.845252	0.576026	0.846821	0.445868	0.750465	0.777407	0.511066
<b>chain_av_results</b>	0.860421	0.570249	0.850136	0.458924	0.750465	0.785391	0.498532
<b>ensemble_results</b>	0.869136	0.577984	0.857531	0.464252	0.750465	0.786273	0.523808
<b>powerset_results</b>	0.819971	0.470123	0.848413	0.488075	0.750465	0.760813	0.464048
<b>BinaryRelevance_results</b>	0.845252	0.576026	0.844802	0.445868	0.750465	0.778328	0.495592

Fig 9: Jaccard similarity score for Epidemic

### 7.5.2 Hamming loss:

Hamming Loss is represented by the equation 4 as  $L_H(y, h(x))$ . Here  $m$  is observed label sets,  $h(x)$  is resulted label which is predicted by classifier,  $(x)$  is 1 for true and 0 for false. Hamming loss determines the ratio of labels that are classified incorrectly. Lower hamming loss value represents higher model accuracy.

$$L_H(y, h(x)) = \frac{1}{m} \sum_{i=1}^m [y_i \neq h_i(x)] \quad (4)$$

hamming	XGBoost	AdaBoost	D-Tree	Naive Bayes	KNN	Random Forest	MLP
<b>OVR_results</b>	0.034928	0.141627	0.043541	0.219378	0.063397	0.051435	0.124641
<b>chain_av_results</b>	0.036435	0.147153	0.044163	0.28567	0.063397	0.053038	0.135622
<b>ensemble_results</b>	0.033493	0.15	0.039713	0.247608	0.063397	0.052632	0.124402
<b>powerset_results</b>	0.053589	0.247129	0.041866	0.190909	0.063397	0.059809	0.144498
<b>BinaryRelevance_results</b>	0.034928	0.141627	0.044498	0.219378	0.063397	0.050957	0.12488

Fig 10: Hamming loss for PProPHET



hamming	XGBoost	AdaBoost	D-Tree	Naive Bayes	KNN	Random Forest	MLP
<b>OVR_results</b>	0.069764	0.181588	0.069257	0.255912	0.120439	0.100338	0.213345
<b>chain_av_results</b>	0.066807	0.191453	0.070963	0.301182	0.120439	0.100422	0.241943
<b>ensemble_results</b>	0.063007	0.189189	0.066892	0.302365	0.120439	0.100169	0.21723
<b>powerset_results</b>	0.088851	0.285811	0.07348	0.280743	0.120439	0.114865	0.26875
<b>BinaryRelevance_results</b>	0.069764	0.181588	0.069595	0.255912	0.120439	0.100338	0.220101

Fig 11: Hamming loss for Epidemic

### 7.5.3 Zero one loss:

Zero-one loss is represented by the equation 5 as  $L_s(y, h(x))$ . Here  $m$  is observed label sets,  $h(x)$  is resulted label which is predicted by classifier,  $(x)$  is 1 for true and 0 for false. Zero-one loss will give the whole predicted output as incorrectly predicted for even one incorrectly predicted label. Lower Zero-one loss value represents higher model accuracy.

$$L_s(y, h(x)) = [y_i \neq h_i(x)] \quad (5)$$

zero-one	XGBoost	AdaBoost	D-Tree	Naive Bayes	KNN	Random Forest	MLP
<b>OVR_results</b>	0.205742	0.748804	0.26555	0.916268	0.30622	0.284689	0.710526
<b>chain_av_results</b>	0.210766	0.737321	0.241627	0.930383	0.30622	0.263876	0.609091
<b>ensemble_results</b>	0.208134	0.73445	0.232057	0.899522	0.30622	0.263158	0.614833
<b>powerset_results</b>	0.758373	0.136364	0.772727	0.284689	0.69378	0.708134	0.370813
<b>BinaryRelevance_results</b>	0.794258	0.251196	0.732057	0.083732	0.69378	0.720096	0.277512

Fig 12: Zero-one loss for PProPHET

zero-one	XGBoost	AdaBoost	D-Tree	Naive Bayes	KNN	Random Forest	MLP
<b>OVR_results</b>	0.375	0.896959	0.378378	0.952703	0.511824	0.494932	0.940878
<b>chain_av_results</b>	0.320946	0.879899	0.344426	0.954054	0.511824	0.458277	0.892568
<b>ensemble_results</b>	0.310811	0.885135	0.336149	0.961149	0.511824	0.45777	0.898649
<b>powerset_results</b>	0.608108	0.087838	0.653716	0.146959	0.488176	0.511824	0.14527
<b>BinaryRelevance_results</b>	0.625	0.103041	0.618243	0.047297	0.488176	0.505068	0.057432

Fig 13: Zero-one loss for Epidemic

#### 7.5.4 F1-score:

F1 score is defined by equation 8. It is the weighted average of recall and precision. The precision and recall score are calculated by  $t_p$ ,  $t_n$ ,  $f_p$ , and  $f_n$ .

- $t_p$ : true positives count.
- $f_p$ : false positives count.
- $t_n$ : true negatives count.
- $f_n$ : false negatives count.

$$Precision_{micro-avg} = \frac{\sum_{i=1}^m t_{pi}}{\sum_{i=1}^m (t_{pi} + f_{pi})} \quad (6)$$

$$Recall_{micro-avg} = \frac{\sum_{i=1}^m t_{pi}}{\sum_{i=1}^m (t_{pi} + f_{ni})} \quad (7)$$

$$F1score_{micro-avg} = 2 \times \frac{2 \times Precision_{micro-avg} \times Recall_{micro-avg}}{Precision_{micro-avg} + Recall_{micro-avg}} \quad (8)$$

<b>F1-score</b>	XGBoost	AdaBoost	D-Tree	Naive Bayes	KNN	Random Forest	MLP
<b>OVR_results</b>	0.940213	0.721018	0.925654	0.521648	0.892669	0.912065	0.773577
<b>chain_av_results</b>	0.937725	0.717159	0.924548	0.55564	0.892669	0.909392	0.765183
<b>ensemble_results</b>	0.943043	0.71409	0.9323	0.602382	0.892669	0.910496	0.786184
<b>powerset_results</b>	0.908347	0.570835	0.928425	0.674286	0.892669	0.898457	0.748752
<b>BinaryRelevance_results</b>	0.940213	0.721018	0.924082	0.521648	0.892669	0.913026	0.771654

Fig 14: F-1 score for PRoPHET

<b>F1-score</b>	XGBoost	AdaBoost	D-Tree	Naive Bayes	KNN	Random Forest	MLP
<b>OVR_results</b>	0.903437	0.720125	0.903756	0.591314	0.834147	0.860235	0.664364
<b>chain_av_results</b>	0.907585	0.711144	0.901159	0.599765	0.834147	0.861036	0.633364
<b>ensemble_results</b>	0.913397	0.72	0.907303	0.606766	0.834147	0.861932	0.661401
<b>powerset_results</b>	0.875943	0.60797	0.898578	0.626013	0.834147	0.841934	0.577199
<b>BinaryRelevance_results</b>	0.903437	0.720125	0.90315	0.591314	0.834147	0.860759	0.647552

Fig 15: F-1 score for Epidemic

## Chapter 8 Results:

The resultant graphs for PRoPHET and Epidemic are compared below:

### 8.1 Jaccard Similarity Score:

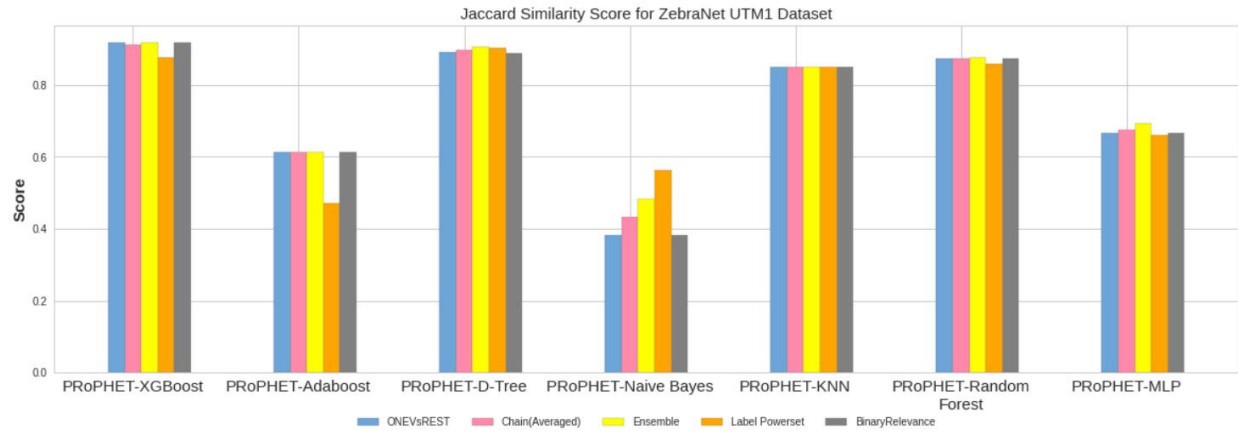


Fig 16: Jaccard similarity score graph for PRoPHET

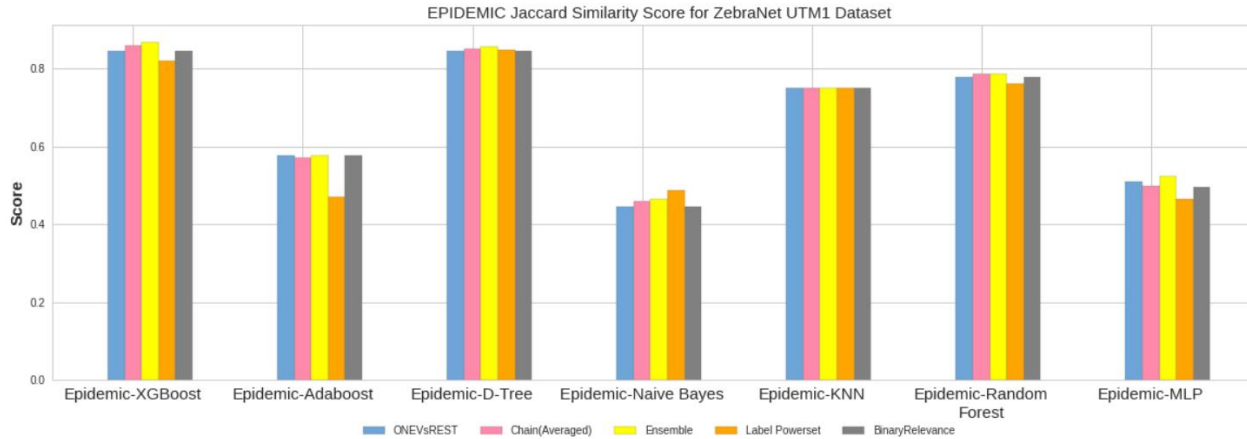


Fig 17: Jaccard similarity score graph for Epidemic

## 8.2 Hamming Loss:

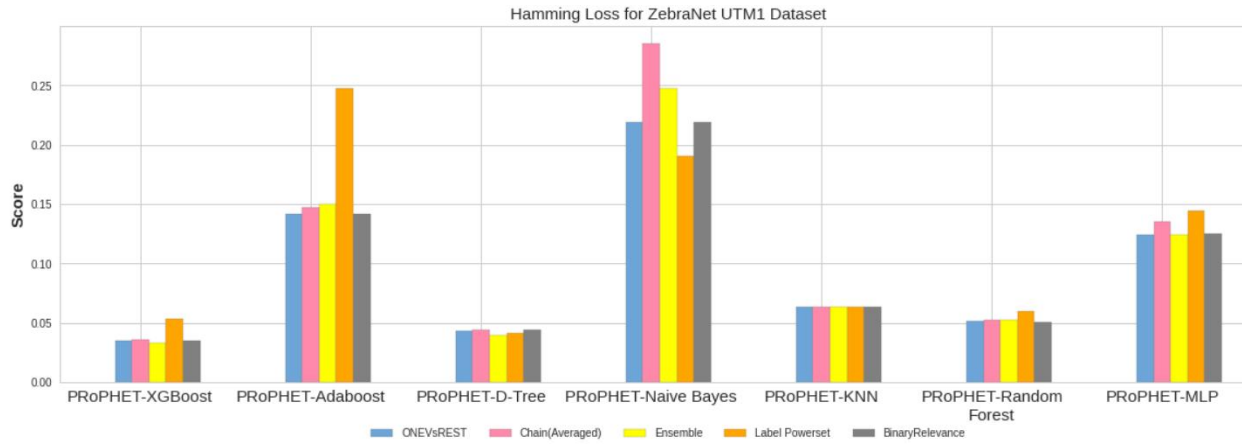


Fig 18: Hamming loss graph for PRoPHET

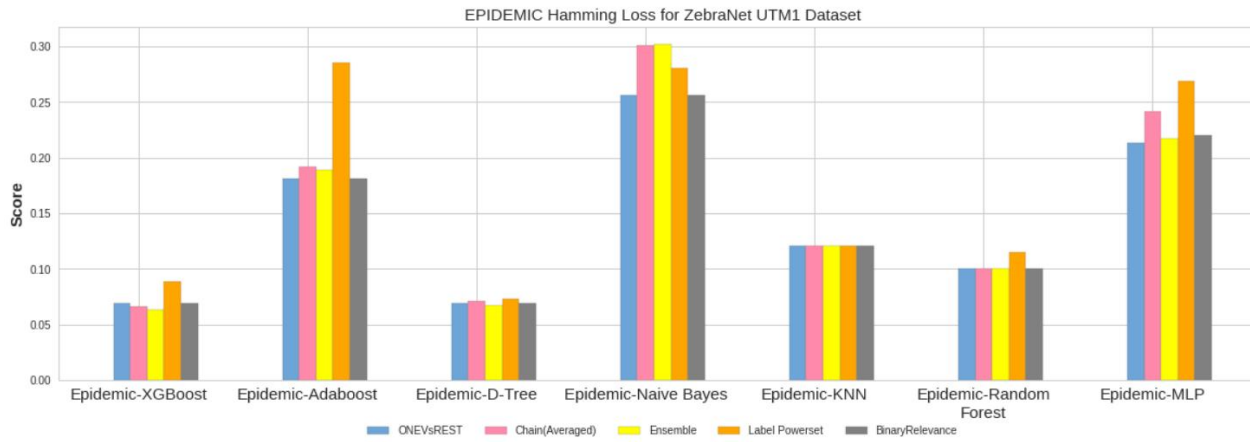


Fig 19: Hamming loss graph for Epidemic

### 8.3 Zero-one

Loss:

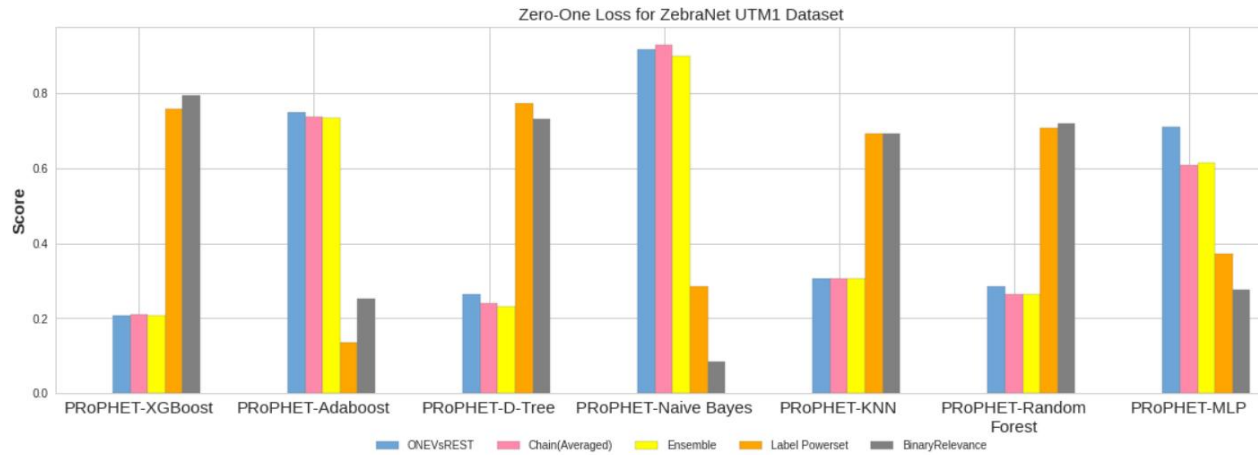


Fig 20: Zero-one loss graph for PRoPHET

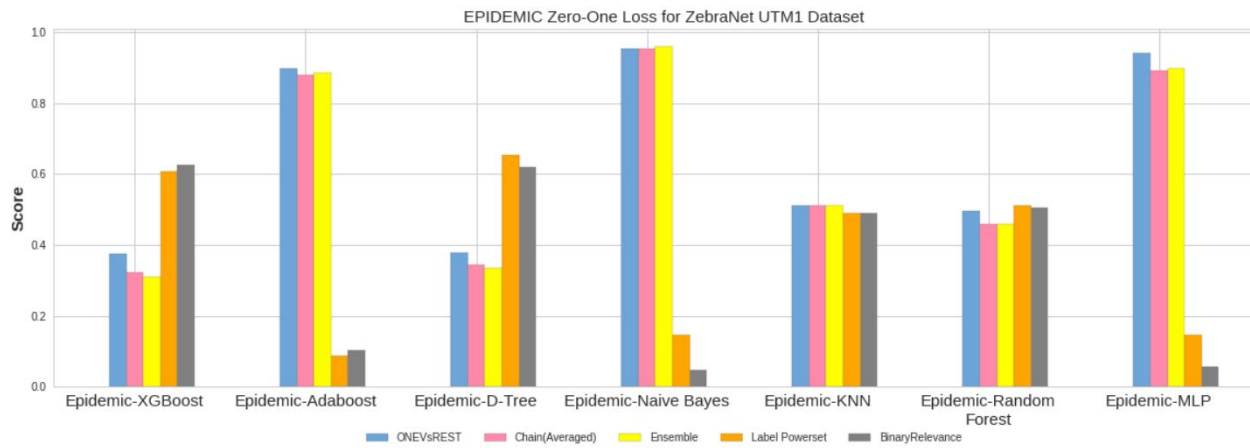


Fig 21: Zero-one loss graph for Epidemic

### 8.4 F1-score:

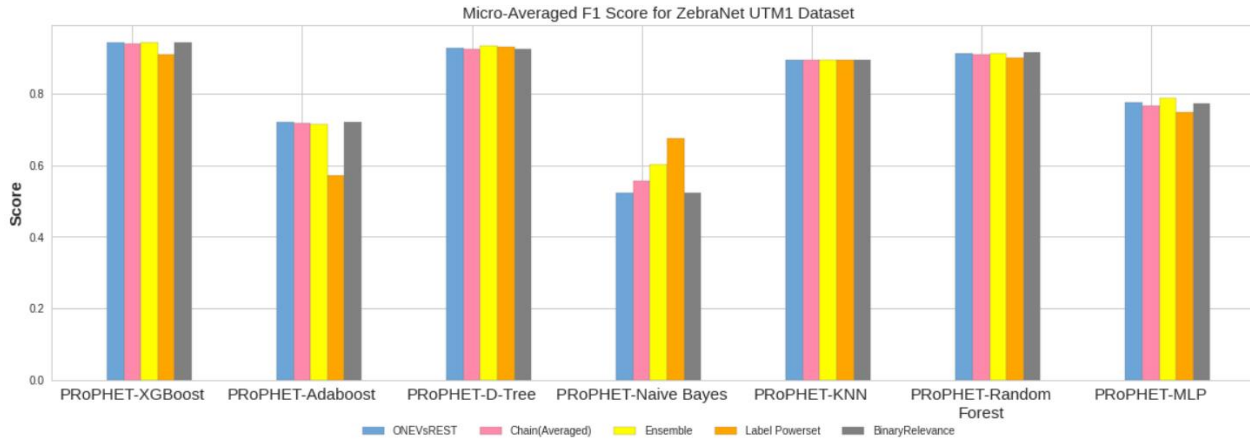


Fig 22: F1-score graph for PRoPHET

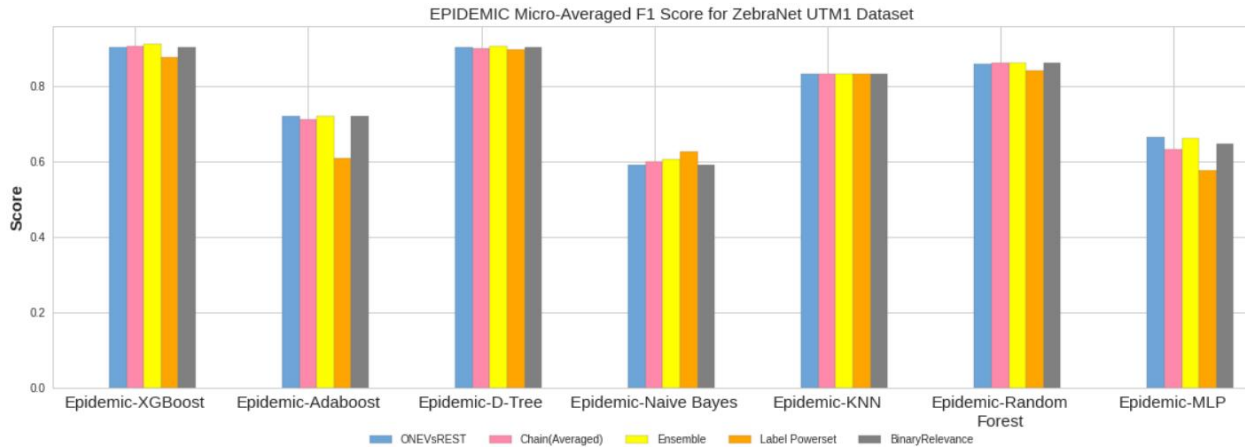


Fig 23: F1-score graph for Epidemic

The results above show that Ensemble technique with XGBoost classifier using prophet gives best result.

Jaccard similarity score = 0.919344 (Ensemble-XGBoost-Prophet) > 0.869136 (Ensemble-XGBoost-Epidemic)

Hamming loss = 0.033493 (Ensemble-XGBoost-Prophet) < 0.063007(Ensemble-XGBoost-Epidemic)

Zero-one loss= 0.208134 (Ensemble-XGBoost-Prophet) > 0.047297(BR- Gaussian NB-Epidemic)

F1-score= 0.943043 (Ensemble-XGBoost-Prophet) > 0.913397(Ensemble-XGBoost-Epidemic)

## **Chapter-9 Conclusion:**

In this work we have two popular DTN routing protocols with the aim to find a suitable node to carry forward the messages in order to decrease the network and increase the message delivery rate. The task of finding the suitable node for transmission of copies is done with the help of machine learning. The task of The Machine Learning techniques used are OneVsRest, Chain classifiers, Ensemble Chain classifiers, Label Power-set transformation and Binary Relevance. The Classifiers used in the work are XGBoost, AdaBoost, kNN, Naïve Bayes, Decision tree, Multilayer perceptron and Random Forest. The results shows that the Ensemble technique with XGBoost classifier as the base classifier gave the best results for PRoPHET as compared to Epidemic. Decision Tree also shows some good results. The parameters are optimized for XGBoost, Random Forest, MLP, and AdaBoost. The value of k in k-NN is set to 1. The number of clusters came out to be 5 as ideal for PRoPHET and Epidemic data. The XGBoost classifier for ensemble based multi-label classification for PRoPHET gave Jaccard score as 0.920819, F-1 score as 0.944785, hamming loss as 0.032297 and zero-one loss as 0.196172.



## References

- [1] Abraham, Abey, and S. Jebapriya. "Routing strategies in delay tolerant networks: a survey." *International Journal of Computer Applications* 42, no. 19 (2012): 0975-8887.
- [2] Cao, Yue, and Zhili Sun. "Routing in delay/disruption tolerant networks: A taxonomy, survey and challenges." *IEEE Communications surveys & tutorials* 15, no. 2 (2012): 654-677.
- [3] D'souza, R. J., and Johny Jose. "Routing approaches in delay tolerant networks: A survey." *International Journal of Computer Applications* 1, no. 17 (2010): 8-14. M. J. Khabbaz, C. M. Assi and W. F. Fawaz, "DisruptionTolerant Networking: A Comprehensive Survey on Recent Developments and Persisting Challenges," in *IEEE Communications Surveys & Tutorials*, vol. 14, no. 2, pp. 607640, Second Quarter 2012.
- [4] M. J. Khabbaz, C. M. Assi and W. F. Fawaz, "DisruptionTolerant Networking: A Comprehensive Survey on Recent Developments and Persisting Challenges," in *IEEE Communications Surveys & Tutorials*, vol. 14, no. 2, pp. 607640, Second Quarter 2012.
- [5] Pöttner, Wolf-Bastian, Felix Büsching, Georg Von Zengen, and Lars Wolf. "Data elevators: Applying the bundle protocol in delay tolerant wireless sensor networks." In *2012 IEEE 9th International Conference on Mobile Ad-Hoc and Sensor Systems (MASS 2012)*, pp. 218-226. IEEE, 2012.
- [6] Shen, Jian, Sangman Moh, and Ilyong Chung. "Routing protocols in delay tolerant networks: A comparative survey." In *ITC-CSCC: International Technical Conference on Circuits Systems, Computers and Communications*, pp. 1577-1580. 2008.
- [7] Sobin, C. C., Vaskar Raychoudhury, Gustavo Marfia, and Ankita Singla. "A survey of routing and data dissemination in delay tolerant networks." *Journal of Network and Computer Applications* 67 (2016): 128-146.
- [8] Zhang, Zhensheng. "Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges." *IEEE Communications Surveys & Tutorials* 8, no. 1 (2006): 24-37.
- [9] Zhu, Ying, Bin Xu, Xinghua Shi, and Yu Wang. "A survey of social-based routing in delay tolerant networks: Positive and negative social effects." *IEEE Communications Surveys & Tutorials* 15, no. 1 (2012): 387-401.

- [10] Sushant, Jain. "Routing in a delay tolerant network." *Proc. of ACM SIGCOMM Computer Communication Review, 2004* (2004): 145-158.
- [11] Tsoumakas, Grigorios, and Ioannis Katakis. "Multi-label classification: An overview." *International Journal of Data Warehousing and Mining (IJDWM)* 3, no. 3 (2007): 1-13.
- [12] Madjarov, Gjorgji, Dragi Kocev, Dejan Gjorgjevikj, and Sašo Džeroski. "An extensive experimental comparison of methods for multi-label learning." *Pattern recognition* 45, no. 9 (2012): 3084-3104.
- [13] Read, Jesse, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. "Classifier chains for multi-label classification." *Machine learning* 85, no. 3 (2011): 333-359.
- [14] Tsoumakas, Grigorios, and Ioannis Katakis. "Multi-label classification: An overview." *International Journal of Data Warehousing and Mining (IJDWM)* 3, no. 3 (2007): 1-13.
- [15] Ahmed, Shabbir, and Salil S. Kanhere. "A bayesian routing framework for delay tolerant networks." In *2010 IEEE Wireless Communication and Networking Conference*, pp. 1-6. IEEE, 2010.
- [16] Portugal-Poma, Lourdes P., Cesar AC Marcondes, Hermes Senger, and Luciana Arantes. "Applying machine learning to reduce overhead in dtn vehicular networks." In *2014 Brazilian Symposium on Computer Networks and Distributed Systems*, pp. 94-102. IEEE, 2014.
- [17] Dudukovich, Rachel, Alan Hylton, and Christos Papachristou. "A machine learning concept for DTN routing." In *2017 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)*, pp. 110-115. IEEE, 2017.
- [18] Boyan, Justin, and Michael Littman. "Packet routing in dynamically changing networks: A reinforcement learning approach." *Advances in neural information processing systems* 6 (1993).
- [19] Valadarsky, Asaf, Michael Schapira, Dafna Shahaf, and Aviv Tamar. "A machine learning approach to routing." *arXiv preprint arXiv:1708.03074* (2017).

- [20] Tsoumakas, Grigorios, and Ioannis Vlahavas. "Random k-labelsets: An ensemble method for multilabel classification." In *European conference on machine learning*, pp. 406-417. Springer, Berlin, Heidelberg, 2007.
- [21] Read, Jesse, Bernhard Pfahringer, and Geoff Holmes. "Multi-label classification using ensembles of pruned sets." In *2008 eighth IEEE international conference on data mining*, pp. 995-1000. IEEE, 2008.
- [22] Read, Jesse. "A pruned problem transformation method for multi-label classification." In *Proc. 2008 New Zealand Computer Science Research Student Conference (NZCSRS 2008)*, vol. 143150, p. 41. 2008.
- [23] G. Tsoumakas, I. Katakis, I. Vlahavas, Effective and efficient multilabel classification in domains with large number of labels, in: *Proceedings of the ECML/PKDD Workshop on Mining Multidimensional Data*, 2008, pp. 30–44
- [24] Zhang, Min-Ling, and Zhi-Hua Zhou. "A review on multi-label learning algorithms." *IEEE transactions on knowledge and data engineering* 26, no. 8 (2013): 1819-1837.
- [25] Santos, A., A. Canuto, and Antonino Feitosa Neto. "A comparative analysis of classification methods to multi-label tasks in different application domains." *Int. J. Comput. Inform. Syst. Indust. Manag. Appl* 3 (2011): 218-227.
- [26] Dudukovich, Rachel, Alan Hylton, and Christos Papachristou. "A machine learning concept for DTN routing." In *2017 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)*, pp. 110-115. IEEE, 2017.
- [27] George, Jean, and R. Santhosh. "Implementation of Machine Learning Classifier for DTN Routing." In *2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, pp. 508-516. IEEE, 2021.
- [28] Dudukovich, Rachel, and Christos Papachristou. "Delay tolerant network routing as a machine learning classification problem." In *2018 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*, pp. 96-103. IEEE, 2018.
- [29] J. Cui, S. Cao, Y. Chang, L. Wu, and Y. Yang, "An adaptive spray and wait routing algorithm based on quality of node in delay tolerant network," *IEEE Access*, vol. 7, pp. 35274–35286, 2019.

- [30] R. W. C. S. Wei Zhihua, L. Dan, and K. Shaoping, "An adaptive spray and wait routing algorithm based on delivery probability of node in dtn," *Journal of Wuhan University: Science Edition*, vol. 65, no. 6, p. 9, 2019.
- [31] H. Hu, D. W. Yang, H. Wang, and J. M. Kuang, "A motion awareness based routing algorithm for delay tolerant network," *Beijing Ligong Daxue Xuebao/Transaction of Beijing Institute of Technology*, vol. 39, no. 1, pp. 68–74, 2019.
- [32] L. Wu, S. Cao, Y. Chen, J. Cui, and Y. Chang, "An adaptive multiple spray-and-wait routing algorithm based on social circles in delay tolerant networks," *Computer Networks*, vol. 189, no. 12, article 107901, 2021.
- [33] K. Liu, Z. Chen, J. Wu, and L. Wang, "Fcns: a fuzzy routing-forwarding algorithm exploiting comprehensive node similarity in opportunistic social networks," *Symmetry*, vol. 10, no. 8, 2018.
- [34] Benhamida, Fatima Zohra, Abdelmadjid Bouabdellah, and Yacine Challal. "Using delay tolerant network for the internet of things: Opportunities and challenges." In 2017 8th International Conference on Information and Communication Systems (ICICS), pp. 252-257. IEEE, 2017.
- [35] <https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d>
- [36] (<https://iq.opengenus.org/gaussian-naive-bayes/>)

## LIST OF PUBLICATIONS

S.NO.	Paper Title, Authors, Conference	STATUS
1	<b>Optimization of Delay Tolerant Routing Using Machine Learning Techniques,</b> Satyam Bajpai, Anamika Chauhan, International Conference on Advances in Computing, Communication Control and Networking (ICAC3N-22)	Accepted
2	<b>Evolution of Machine Learning Techniques for Optimizing Delay Tolerant Routing</b> Satyam Bajpai, Anamika Chauhan, International Conference on Advances in Computing, Communication Control and Networking (ICAC3N-22)	Accepted