Received: May 1, 2020. Revised: June 9, 2020.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Socialized Proficient Routing in Opportunistic Mobile Network Using Machine Learning Techniques

Vimitha Rajendran Vidhya Lakshmi[1] **Gireesh Kumar Thonnuthodi**[1]*

[1]*TIFAC-CORE in Cyber Security,*
Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, India
* Corresponding author's Email: t_gireeshkumar@cb.amrita.edu

Abstract: In Opportunistic Mobile Network, routing remains as a challenging issue since participating nodes are strangers to each other and are not trustworthy. An efficient routing model entitled Socialized Proficient Routing (SPR) using Machine Learning (ML) technique is proposed in this paper. In SPR, the relay nodes are selected based on human-social characteristic of the nodes, in-order to attain high trustworthiness. SPR model embodies three phases. In feature selection phase, the significant features are extracted from the training dataset using Boruta wrapper algorithm. Naïve-Bayes, Decision-Tree, Neural-Networks, Support-Vector-Machine, and Random-Forest (RF) are the different ML classifiers used in the training phase. Testing phase accurately selects the trusty neighbour (friendship) nodes for routing. This model is investigated over MIT reality mining dataset and is evaluated using Opportunistic Network Environment simulator. Experimental results prove that SPR_RF performs the best among the classifiers with 0.93 Message-Delivery-Probability, 894.91s Average-Delivery-Delay, 3.08 Average-Hop-Count, Zero Dropped-Message and 45.15 Overhead-Ratio.

Keywords: Routing, Machine learning, Mobile networks, Opportunistic networks.

1. Introduction

Opportunistic Mobile Network (OMN) is a form of Delay Tolerant Network (DTN) [1] and are known to be the future Mobile Social Networks [2]. These networks are infrastructure-less and no communication path exists between source and destination nodes. During data transfer, DTNs first search for an end-to-end path. If such a path is absent, the data is forwarded opportunistically. OMNs on the other hand always transfer data opportunistically.

While routing, OMNs selects the neighbour nodes based on previous history or human-social characteristics of the nodes. Human-social characteristics include: meeting duration, common interests, similar communities [3], centrality [4], friendship strength [5] and call duration. These features define the trustworthiness and closeness of mobile nodes participating in such kind of networks. When OMNs use the social characteristics of a node as relay node selection criteria during routing, they become Opportunistic Mobile Social Networks (OMSNs) [6]. An important research area in OMSN is routing. Although there are many researches on this topic, efficient and secured routing remains as a major issue. For proficient routing, proper selection of the intermediate nodes is important since routing occurs opportunistically. This helps to reduce delivery latency, increase trust and delivery probability.

Depending on the type of casting, OMN routing methods can be classified into three categories: unicasting [7], unlimited multicasting [8] and limited multicasting [3, 4, 9, 10]. In unicasting, delivery latency is very high and delivery ratio is very low. The solution for these drawbacks came in the form of unlimited multicasting. Here, flooding of messages occurs since the total number of packets increase exponentially each time when a node encounters its relay node. This process increases delivery probability and decreases delivery latency, but resource and network overhead are considered very high. Therefore, in-order to solve the drawbacks of both unicasting and unlimited multicasting, a new approach emerged known as limited multicasting.

Limited multicasting reduces the flooding of the messages and message replicas by reducing the number of relay nodes to which a particular message should be transmitted. Thus, selection of relay nodes should be controlled by some parameters to limit the message replicas. These controlling parameters comprise of history or social characteristics of the mobile nodes. Social characteristics of the mobile nodes are considered in-order to increase the trust between the nodes.

Machine Learning (ML) techniques are used to train the mobile nodes with these social characteristics in-order to select the best and accurate relay nodes while routing. There exist few routing models based on different ML techniques [5, 11-13] but no comparative analysis has been carried out between them. And the number of controlling parameters also varies in each work. There are no conditions or guidelines to pick such control parameters. More control parameters do not improve the efficiency of the model. In this way, legitimate choice and utilization of these controlling parameters just as appropriate determination of relay nodes still stays as an open problem in OMSNs.

This paper presents a new and efficient model for relay node selection in OMSNs. This model is referred as Socialized Proficient Routing (SPR). SPR comprises of three phases: Significant Feature Selection Phase, Training Phase and Testing Phase, to select significant features, to train the classifiers, and to accurately route the data respectively. The different ML techniques used in SPR are evaluated using performance metrics like Message Delivery Probability (MDP), Average Delivery Delay (ADD), Average Hop Cost/Count (AHC), Dropped Messages (DM) and Overhead Ratio (OR). Finally, the best ML technique is selected to accomplish proficient routing in OMSNs.

The main contributions of this work are:

1) Utilization of the feature selection phase, which extracts the significant features from the original dataset. Existing related works make use of the whole dataset for routing decisions. This increases the overall complex nature of the routing model. Significant features alone are sufficient to design an efficient routing model. These significant features are considered to have great impact on deciding the friendship between these mobile nodes.

2) Performs comparative analysis between different ML classifiers and finally selects the classifier with better performance outcome to design the routing model.

This paper is organized as follows. The related works are presented in Section 2. In Section 3, the proposed SPR model is presented. Section 4 explains the experimental setup. The results and analysis are presented in Section 5. Finally, we conclude in Section 6 mentioning some future directions of research on the topic presented in this paper.

2. Related works

2.1 Routing methods

Direct Delivery [7] is a type of unicasting routing method where the source node holds the message until it meets the destination node. The message is transmitted only to the destination node. The advantage of this routing method is low network overhead because nodes maintain only single copy of the messages. Limitations: High delivery latency and low delivery ratio since if no nodes come in contact, no transferring is done.

Probabilistic Routing Protocol using History of Encounters and Transitivity (PRoPHET) [8] is an unlimited multicasting routing method. In PRoPHET each node maintains a summary vector as well as a delivery predictability metric. Depending on these two metrics, message hops takes place. Low delivery latency and high delivery ratio are achieved by this method since multiple copy of messages are generated within the network. Limitations: High network overhead since number of message replicas are high.

Limited multicasting protocols overcome the limitations of unicasting and unlimited multicasting. Spray and Wait [9] protocol control the level of flooding. Spray phase spreads the message to the relay nodes, while in the Wait phase, if the destination is not found during Spray phase, then relay nodes having the copy of the message transfers the message directly to the destination node. Some protocols like Social-aware Content-based Opportunistic Routing Protocol (SCORP) [3] and Multi-Layer Social network based Opportunistic Routing (ML-SOR) [4] utilize social characteristics of the nodes to select relay nodes.

2.2 Routing methods using ML techniques

kROp [11] uses an optimized K-means clustering algorithm. This method considers the features such as the encounter history of the node, distance of the node from the destination node, buffer space remaining and the number of messages delivered successfully. These features are extracted, clustered and optimized in-order to find the accurate relay node. MLProph [12] is an improvement to PROPHET+ [14] protocol. The ML techniques used are neural network and decision tree model. MLProph is trained with twelve features using these ML techniques and the next hop node selections are made. FSF [5] is based on friendship and selfishness forwarding. The ML technique used is a Naïve Bayes classifier which classifies the friendship nodes and messages are forwarded only to these selected nodes. Selfishness of a node is calculated based on the node's reputation value. The features considered in FSF are the meeting frequency, contact duration, call amount and amount of text messages.

3. Proposed model

A Socialized Proficient Routing (SPR) model is proposed in this paper. Mobile nodes are classified into two categories (friendship nodes and stranger nodes) based on friendship between them. Only friendship nodes are selected as relay nodes for forwarding the message from one node to another. The three phases of the proposed model are: Significant Feature Selection phase, Training phase, and Testing phase. Fig. 1. illustrates the system model of the proposed work.

3.1 Dataset description

The dataset used in this work is MIT reality mining dataset [15]. The reason for choosing this dataset is the availability of the friendship attribute and also the usage of this dataset by similar method [5]. This dataset consists of data collected from 94 nodes over time period of ten months. The raw dataset includes attributes like meeting frequency, contact duration, total calls, total messages and a response attribute; friendship. The raw dataset consists of 8680 instances.

3.2 Significant feature selection phase

This phase selects significant features from the whole raw dataset. Feature selection speeds up the training phase of the ML classification algorithms, reduces the complexity of the model by decreasing the computational cost (execution time) without compromising the accuracy of the model. In SPR, Boruta wrapper algorithm [16] is used to select significant features. This algorithm acts as a wrapper around Random Forest (RF). Here, attribute-importance is evaluated by Z-score, as this measure gives mean accuracy loss of the classifier. Boruta algorithm outperforms other traditional wrapper algorithms [17], as it considers all-relevant feature selection rather than considering minimal optimal methods. The above phase is implemented in R [18] to select the significant features.

The graphical representation depicting the importance of features is shown in Fig. 2. Table 1

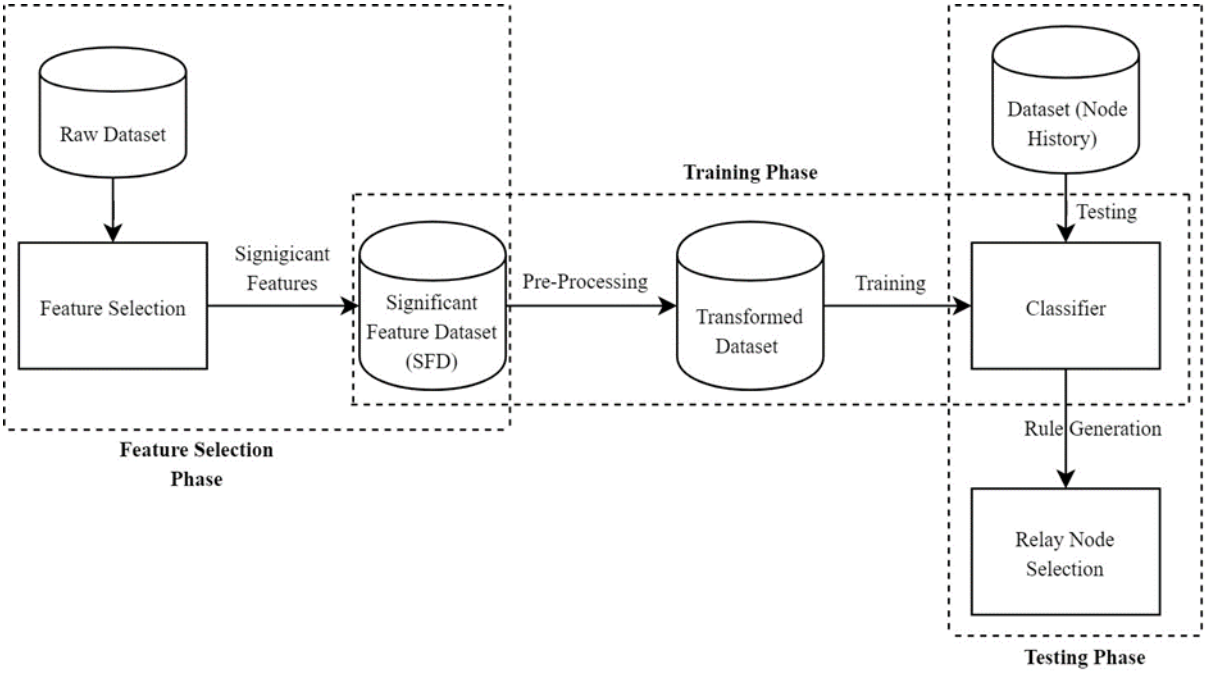


Figure 1. System model of the proposed work (SPR)

Received: May 1, 2020. Revised: June 9, 2020.

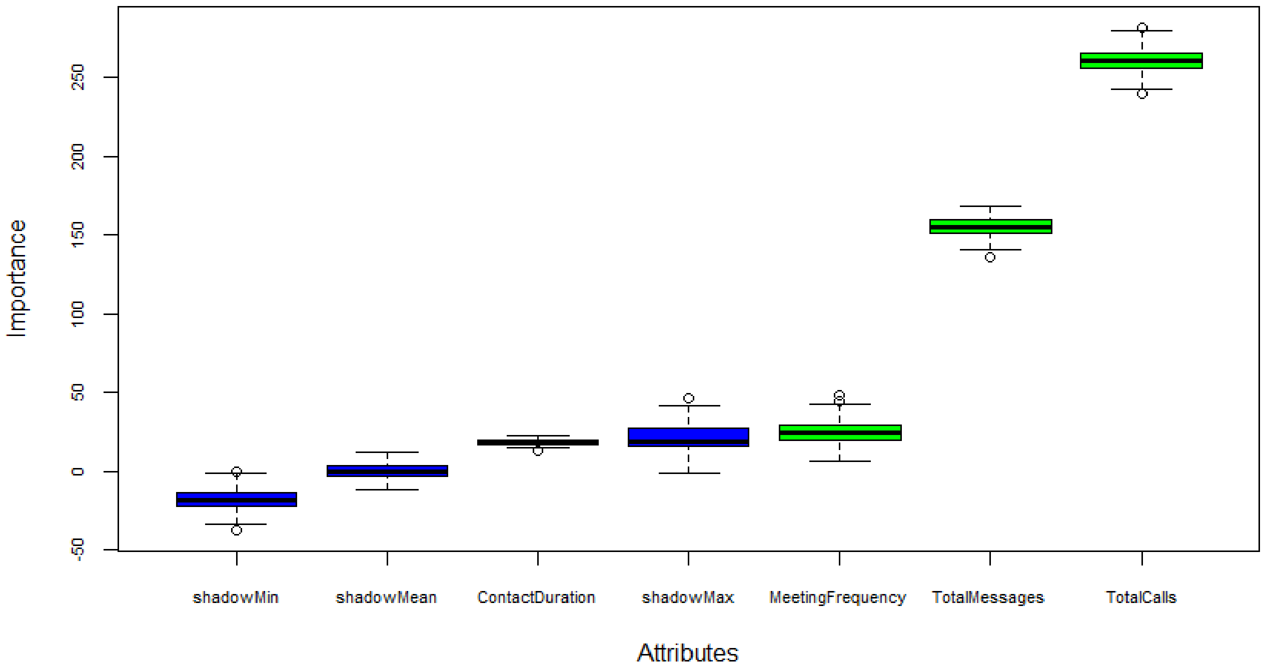


Figure. 2 Feature selection for the raw dataset (Significant features are highlighted in green colour)

Table 1. Original versus selected significant features

Total Attributes (4):	Meeting Frequency, Contact Duration, Total Calls, Total Messages
Significant Attributes (3):	Meeting Frequency, Total Calls, Total Messages

Table 2. Category and range values in T-SFD

Attributes	Value Range	Category
Meeting Frequency	0-4 5-15 16-50	Low Medium High
Total Calls	0-2 3-5	Low High
Total Messages	0-2 3-5	Low High

intimates the original features versus selected significant features of the dataset, obtained as a result of applying Boruta algorithm in R. The output of this phase will be a dataset with significant features and a response variable named as Significant Feature Dataset (SFD).

3.3 Training phase

SFD is pre-processed into transformed dataset (T-SFD) by categorizing the significant features into different ranges like low, medium and high. For example, if the meeting frequency is in the range 0-50, then 0-4 is categorized into low, 5-15 as medium and 16-50 as high. Similarly, all the attributes in SFD are categorized. Table 2 reveals the T-SFD range values after pre-processing the SFD. This T-SFD is used to train the different ML classifiers [19]. In SPR, one classifier from each category is selected, to conclude which classifier performs the best of all the other classifiers. Table 3 depicts the classifier and the category of the classifier.

Table 3. Classifier and its category

Classifier	Category
Naïve Bayes	Probabilistic Classifier
Decision Tree	Tree Based Classifier
Neural Networks	Bio-Inspired Classifier
Support Vector Machine	Binary Kernel Based Classifier
Random Forest	Ensemble Classifier

3.3.1 Naïve Bayes classifier (NB)

If X is an instance vector containing the 3 features of T-SFD, then $X= (x_1, x_2, x_3)$ where $x_1=$ Meeting Frequency, $x_2=$ Total Calls and $x_3=$ Total Messages. Bayesian classifier is used to find out X belongs to which class (there are two classes: Friendship class f and Stranger Class s). X belongs to the class with maximum probability. The mathematical expression of this model is as follows:

$$P\left({}^{f}/_{X}\right) = P(f) \times \prod_{i=1}^{3} P\left({}^{x_i}/_{f}\right) \qquad (1)$$

$$P\left({}^{s}/_{X}\right) = P(s) \times \prod_{i=1}^{3} P\left({}^{x_i}/_{s}\right) \qquad (2)$$

If the value of (1) > (2), then X belongs to friendship class else X belongs to stranger class.

3.3.2 Decision Tree classifier (DT)

This classifier uses C4.5 (J48) algorithm, to generate decision tree. In C4.5 the splitting of the training data depends on the value of normalized information gain, known as split information, and gain ratio. If T is the training dataset T-SFD with 3 attributes and 2 classes then,

$$Entropy(T) = -\sum_{i=1}^{2} p_i \, log_2 \, p_i \tag{3}$$

where p_i is the fraction of samples in the class i. Each attribute A have b sub-partitions then,

$$Entropy(T, A) = \sum_{j=1}^{b} P(A_j) \times Entropy(A_j) \tag{4}$$

The equations for calculating information gain, spilt information and gain ratio are as follows:

$$Gain(T, A) = Entropy(T) - Entropy(T, A) \tag{5}$$

$$SpiltInfo(T, A) = -\sum_{j=1}^{b} P(A_j) \, log_2 \, P(A_j) \tag{6}$$

$$GainRatio(A) = \frac{Gain(T,A)}{SpiltInfo(T,A)} \tag{7}$$

3.3.3 Neural Network classifier (NN)

In this classifier, a type of feedforward neural network called Multi-Layer Perceptron (MLP) is used. Let $\{x_1, x_2, x_3\}$ be the input attributes where x_1= Meeting Frequency, x_2= Total Calls and x_3= Total Messages and randomly assigned weights be $\{w_1, w_2, w_3\}$, $\boldsymbol{f}$ be the activation function (sigmoid function), $\{h_1, h_2, ..., h_m\}$ be the nodes in the hidden layer. The output of each node in the hidden layer is:

$$h_i = \boldsymbol{f}(x_1 w_1 + x_2 w_2 + x_3 w_3) \quad \text{i=1, 2, ..., m} \tag{8}$$

The error is calculated at the output layer using Least Mean Square algorithm. Backpropagation is done and the weights are adjusted to minimize the error at the output layer. If X is the actual output and Y is the predicted output, then error E is:

$$E = \frac{1}{2}\sum(X - Y)^2 \tag{9}$$

If n is the learning rate, then change in weight w is:

$$\delta(w) = -n(\frac{dE}{dx}) \tag{10}$$

Updated weight is:

$$w_{new} = \delta(w) + w_{old} \tag{11}$$

3.3.4 Support vector machine classifier (SVM)

This classifier is well suited for extreme cases of data points, for precise classification. The input, X consists of 3 features, X= $\{x_1, x_2, x_3\}$, where x_1= Meeting Frequency, x_2= Total Calls, x_3= Total Messages and output y (friendship class or stranger class). Let w be the weight vector, D the datapoints, D = {(X_1, y_1), (X_2, y_2), ... (X_n, y_n)} for n instances and b the bias (scalar value). Then the hyperplane is written as:

$$w \times D + b = 0 \tag{12}$$

The support vectors (data points) that fall on the hyperplane is selected as:

$$\forall(X_i, y_i): y_i(w \times X_i + b \geq 1) \tag{13}$$

where,

$$w = \sum \alpha_i \times X_i, \quad \alpha_\text{i} \text{ is Langlier's multiplier} \tag{14}$$

$$b = y_\text{c} - w \times D_\text{c} \text{ for any } D\text{c}: \alpha c \neq 0 \tag{15}$$

Classification is finally done with the decision boundary:

$$f(x) = \sum_{i=1}^{l} y_i \alpha_i D_i \times D + b \tag{16}$$

Where l is the total number of support vectors.

3.3.5 Random forest classifier (RF)

Collection of decision trees are aggregated to form a RF. As the number of trees in RF gets higher, the classification accuracy also gets higher. In this classifier bootstrapped datasets are used to create each individual decision tree. Each individual decision tree grows to a maximum depth without pruning. If there are M features in bootstrap dataset, then m features are randomly selected from M such that $m < M$. The decision tree is built with m features by choosing the best spilt (information gain) at each step.

3.4 Testing phase

Initially, the nodes in the network are allowed to warm-up for a short period of time. The nodes collect each other's history in this time period. After collecting the history of the nodes, the trained dataset tests for the correct classification of friendship nodes. The result of the testing phase will be the classification of the contacting relay nodes as friendship node or stranger node. The data is passed to the relay node only if the classifier determines the relay node as friendship node, else the data will not be routed. Thus, routing happens only between friendship nodes in the network. This procedure increases the overall trust between the nodes in the network.

4. Experimental settings

Opportunistic Network Environment (ONE) simulator [20] is used to evaluate the performance of the proposed model; SPR. For simulation the network environment has to be configured with the point of communication (interface) between the nodes. Among the list of interfaces such as Bluetooth, P2P link and Ethernet, Bluetooth interface is used for node communication with transmission speed (data rate through the Bluetooth interface) of 250kBps and transmission range of 50m. Common settings for the ONE simulator are conveyed in Table 4. The following performance metrics are used to evaluate the performance of SPR by varying Number of Nodes, Buffer Size, Message Time-To-Live (TTL) and Message Generation Interval.

MDP: Rate of successful delivery of messages to their specified destination nodes. ADD: Average time interval taken by the message to travel from source node to the destination node (in seconds). AHC: Average number of relay nodes through which a particular message passes through, starting from the source node to the destination node. DM: Number of messages that fail to reach the destination node. OR: Average number of message replicas of a particular message. This metric measures the network overhead. Low AHC leads to low OR. A good classifier will have high MDP, low ADD, low AHC, zero DM and low OR.

Table 4. ONE simulator settings

Movement Model	ShortestPathMapBasedMovement
Simulation Time	10,000s
Wait Time (s)	0-120
Movement Speed (m/s)	0.5-1.5
Message Size	500kB-1MB
Warm up Time (s)	1000
World Size (m)	4500 × 3400 (width × height)

The algorithm for the proposed model is as follows:

W→ Warm up time period for the nodes (in seconds).
C → Set of all five machine learning classifiers. {1: NB, 2: DT, 3: NN, 4: SVM, 5: RF}
T → T-SFD (Training Dataset)
S → Data Sender Node *R*→ Data Receiver Node
P → Performance Metrics {1: MDP, 2: ADD, 3: AHC, 4: DM, 5: OR}
V → Varying Parameters {1: Number of nodes, 2: Buffer Size, 3: Time-To-Live, 4: Message Generation Interval}

```
Algorithm SPR (W, C, T, S, R, P, V)
    1.  For w=0 to W
    2.      Collect history of the nodes
    3.  For C=1 to 5
    4.  Train the classifier with T
    5.  Test the classifier during routing
    6.      If (Friendship (S, R) == True)
    7.          Classify as Friendship nodes
    8.          Select R as relay node
    9.          Data is routed to R
    10.     Else
    11.         Classify as Stranger Nodes
    12.         Data in stored in buffer(S)
    13.         Data is not routed to R
    14. For P=1 to 5
    15.     For V=1 to 4
    16.         Performance is computed
    17. Compare the values obtained in step 16.
    18. Select the best classifier
```

5. Results and analysis

5.1 Varying number of nodes

The number of nodes is the total number of mobile nodes inside the simulation area. It is varied from 10 to 50. In addition to Table 4, the ONE simulator default settings for this analysis are i) Buffer Size: 50MB, ii) Message TTL: 300 minutes and iii) Message Generation Interval: 150-200s. The graphical representation is manifested in Fig. 3.

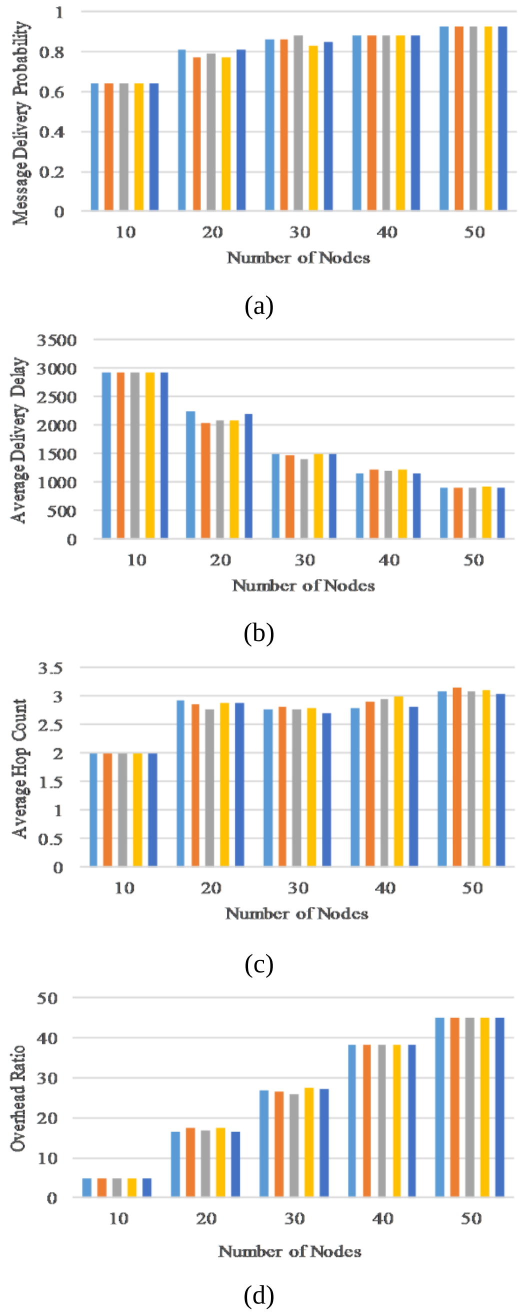


Figure. 3 Graphical representation of results obtained by varying number of nodes: (a) number of nodes vs MDP, (b) number of nodes vs ADD (in seconds), (c) number of nodes vs AHC, and (d) number of nodes vs OR

From Fig. 3. it is clear that as the number of nodes increases from 10 to 50, the MDP increases (Fig. 3(a)) (number of nodes α MDP) and ADD decreases (Fig. 3(b)) (number of nodes α 1/ADD). AHC is directly proportional to the number of relay nodes, thus increase in AHC is due to the raise in relay nodes when there exist more nodes in the network (Fig. 3(c)). Similarly, increase in number of relay nodes results in proliferation of message

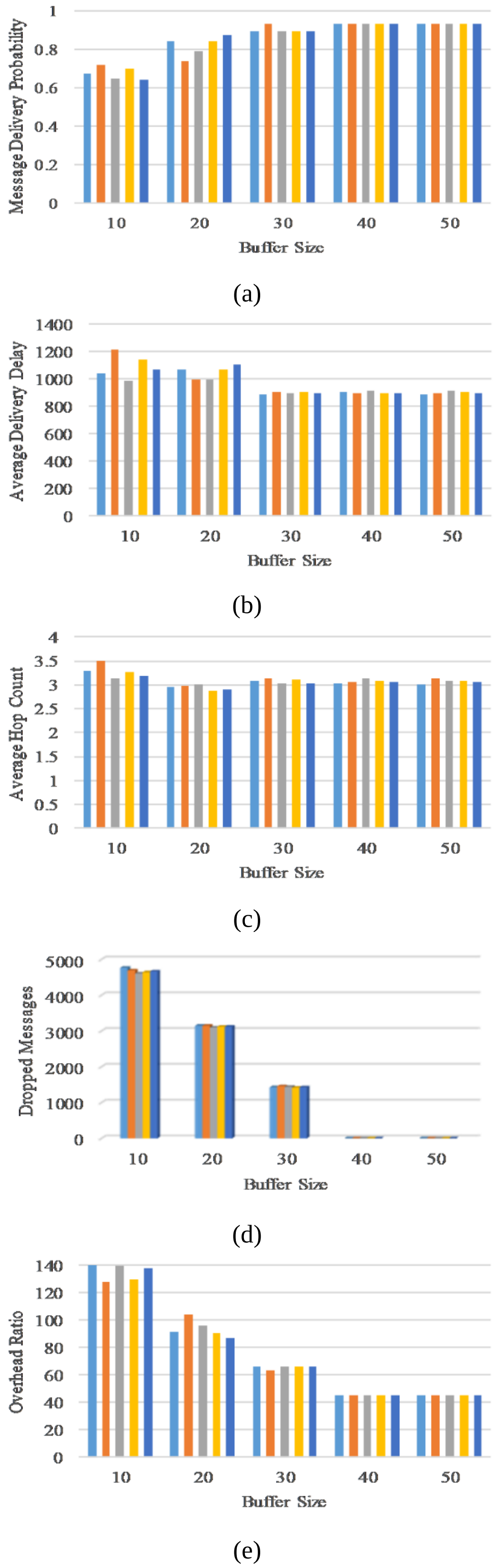


Figure. 4 Graphical representation of results obtained by varying buffer size: (a) buffer size vs MDP, (b) buffer size vs ADD (in seconds), (c) buffer size vs AHC, (d) buffer size vs DM, and (e) buffer size vs OR

replicas, which results in high network overhead (Fig. 3(d)). It is inferred that when there are more mobile nodes in OMSN, the network will have high MDP and low ADD while routing. A good routing method must exhibit high MDP and low ADD. The number of DM are maintained at zero. The optimal results are obtained when the number of nodes is high.

5.2 Varying buffer size

The buffer size is the capacity of a mobile node to store the messages received and forwarded. It is varied from 10 to 50MB. In addition to Table 4, the ONE simulator default settings for this analysis are i) Number of Nodes: 50, ii) Message TTL: 300 From Fig. 3. it is clear that as the number of nodes increases from 10 to 50, the MDP increases (Fig. 3(a)) (number of nodes α MDP) and ADD decreases (Fig. 3(b)) (number of nodes α 1/ADD). AHC is directly proportional to the number of relay nodes, thus increase in AHC is due to the raise in relay nodes when there exist more nodes in the network (Fig. 3(c)). Similarly, increase in number of relay nodes results in proliferation of message replicas, which results in high network overhead (Fig. 3(d)). It is inferred that when there are more mobile nodes in OMSN, the network will have high MDP and low ADD while routing. A good routing method must exhibit high MDP and low ADD. The number of DM are maintained at zero. The optimal results are obtained when the number of nodes is high.

5.3 Varying buffer size

The buffer size is the capacity of a mobile node to store the messages received and forwarded. It is varied from 10 to 50MB. In addition to Table 4, the ONE simulator default settings for this analysis are i) Number of Nodes: 50, ii) Message TTL: 300 minutes and iii) Message Generation Interval: 150-200s. The graphical representation is manifested in Fig. 4.

From Fig. 4. it is clear that as the buffer size increases, MDP increases (Fig. 4(a)), because high capacity buffers can hold additional messages while routing. Increase in buffer size allot space for new messages without dropping the old ones. ADD and AHC decreases (Fig. 4(b) and 4(c) respectively), since messages reach their destination in less time and with fewer relay nodes. The count of dropped messages will be larger for smaller buffer size. As seen in Fig. 4(d) when the buffer size pass over 40MB, the number of DM prolong to Zero.

Fewer relay nodes result in low proliferation of message replicas which decreases OR (Fig. 4(e)). The optimal results are obtained when the buffer size is high. Thus, it is inferred that a good routing method should have larger buffer capacity.

5.4 Varying message Time-To-Live (TTL)

The message TTL indicates the lifespan of a particular message. After this timespan the message is discarded. It is varied from 100 to 500 minutes. In addition to Table 4, the ONE simulator default settings for this analysis are i) Number of Nodes: 50, ii) Buffer Size: 50MB and iii) Message Generation Interval: 150-200s. The graphical representation is manifested in Fig. 5.

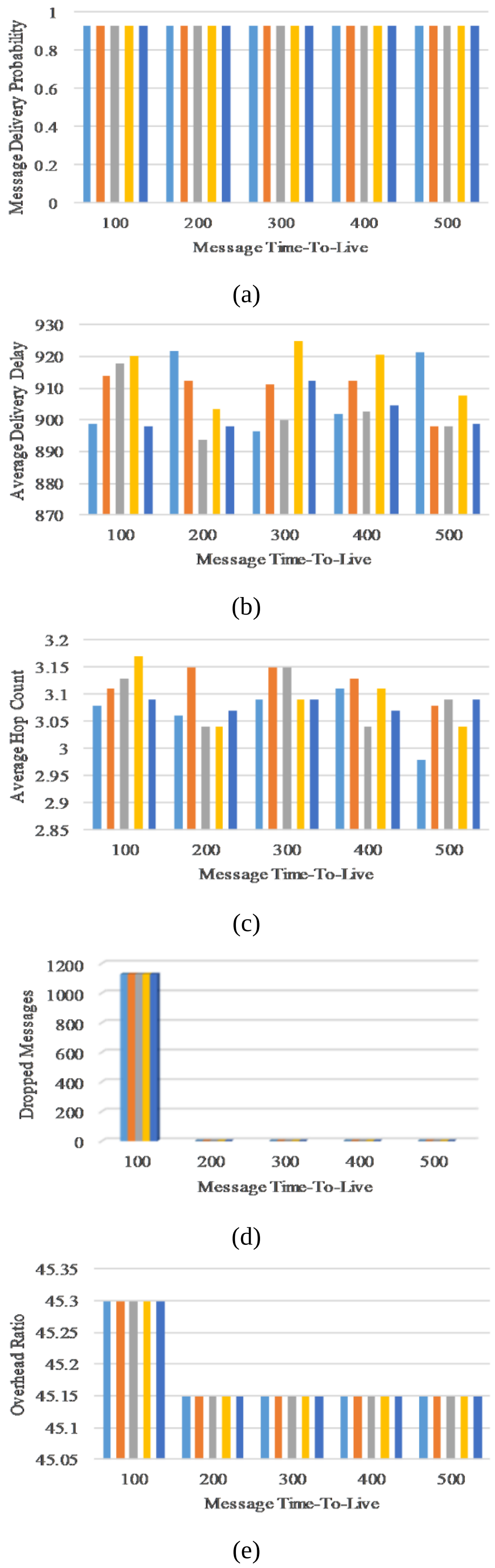
From Fig. 5. it is clear that as the message TTL is varied from 100 to 500 minutes, the MDP remains stable (Fig. 5(a)). This is because all the messages reach the destination within the first few minutes for this particular environmental setting. Longer lifespan of the messages is crucial in order to increase the MDP without the message being dropped out from the buffer. The stochastic nature of ADD values in Fig. 5(b) indicates that, the message TTL does not play a significant role in ADD. The message is transmitted to minimum number of relay nodes, when a message stays for longer time in a node’s buffer. This leads to low AHC (Fig. 5(c)).

The number of DM is high for small message TTL, since all the messages are held in buffer for a short period of time. As in Fig. 5(d) when message TTL passes over 200 minutes, the number of DM prolong to zero. Lower AHC results in low rate of message replica, which decreases the OR (Fig. 5(e)). The optimal results are obtained when message TTL is high. It is inferred that, high message TTL contributes a good routing method with high MDP, low AHC, low DM and low OR.

5.5 Varying message generation interval

The message generation interval is varied from 0-50s to 200-250s. A new message is created within this specified interval. In addition to Table 4, the ONE simulator default settings for this analysis are i) Number of Nodes: 50, ii) Message TTL: 300 minutes and iii) Buffer Size: 50MB. The graphical representation is manifested in Fig. 6.

From Fig. 6. it is clear that as the message generation interval increases, MDP first increases till 150-200s and then decreases (Fig. 6(a)). If generation interval is too low then, more messages are created. The chances for some messages to be dropped is high which results in low MDP. If generation interval is too high then, less messages



(a)

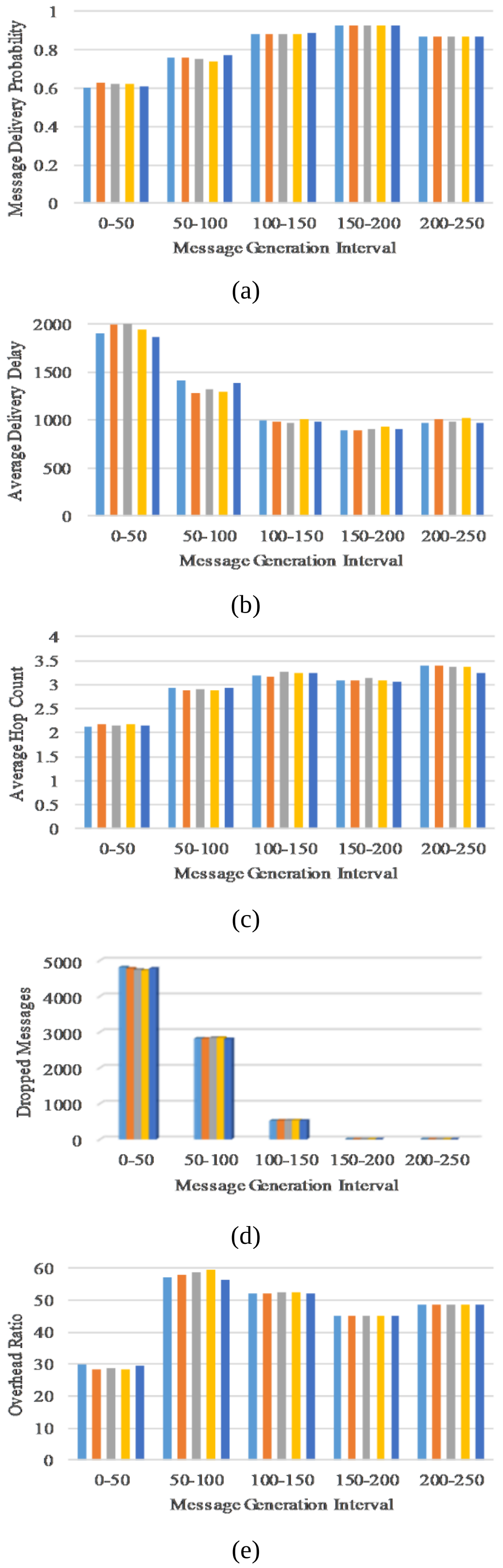
(b)

(c)

(d)

(e)

Figure 5. Graphical representation of results obtained by varying message TTL: (a) message TTL vs MDP, (b) message TTL vs ADD (in seconds), (c) message TTL vs AHC, (d) message TTL vs DM, and (e) message TTL vs OR



(a)

(b)

(c)

(d)

(e)

Figure 6. Graphical representation of results obtained by varying message generation interval: (a), message generation interval vs MDP, (b) message generation interval vs ADD (in seconds), (c) message generation interval vs AHC, (d) message generation interval vs DM, and (e) message generation interval vs OR

are created. The chances for these messages to reach their destination nodes is low which results in low MDP. Whereas, ADD first decreases till 150-200s and then increases, as the message generation interval increases (Fig. 6(b)). When there more messages, all the messages cannot be transmitted to another node during the short contact period between two nodes. Similarly, when there are less messages, chances of meeting their suitable relay nodes will be low. Therefore, ADD will be high.

The stochastic nature of AHC values in Fig. 6(c) indicates that, the message generation interval does not play a significant role in AHC. Larger message generation interval leads to smaller number of messages to create which in turn decrease the number of DM. The DM count is high for small message generation interval and from interval 150-200s the number of DM prolong to zero (Fig. 6(d)). More message generation leads to proliferation of message replicas which increases OR. Thus, as the message generation interval increases, OR also increases (Fig. 6(e)). It is inferred that, for a routing protocol with optimal results, the message generation interval should be neither too high nor too low.

Overall Inference: From Fig. 3-6 it is analyzed that, among the 5 classifiers (NB, DT, NN, SVM, RF), comparatively RF performs the best in terms of all the performance evaluation parameters. For example, when number of nodes is 50 and for the ONE simulator settings described in section 5.1. For all the 5 classifiers: MDP is 0.93, DM is 0 and OR is 45.15. ADD is 909.48s, 913.89s, 910.93s, 939.12s, 894.91s and AHC is 3.09, 3.16, 3.09, 3.11, 3.08 for NB, DT, NN, SVM, RF respectively. Similarly, it is identified that, RF performs better for all the other varying parameters.

5.6 Comparison with existing techniques

Existing routing methods are divided into two categories: Non-ML techniques and ML techniques. From non-ML techniques, routing methods from each casting type is selected. Limited multicasting is more focused, since it overcomes the limitations of unicasting and unlimited multicasting. The controlling parameters are chosen to have social characteristics, due to the presence of social attributes, which increases trust between the nodes in the network. Furthermore, ML techniques improves the evaluation parameters of the routing method. Thus, many recent works focus on ML-based routing method.

SPR model without feature selection phase is denoted as SR (Socialized Routing). SR model is designed in-order to highlight the importance of feature selection phase. SPR is compared with other existing techniques and the results obtained are summarized in Table 5. The ONE simulator settings

Table 5. Comparison with existing techniques

Category	**Casting Type**	**Controlling Parameters**	**Routing Protocol**	**MDP**	**ADD**	**AHC**	**DM**	**OR**
Non-ML Techniques	Unicasting	None	Direct Delivery [7]	0.44	2617.08	1.00	0	00.00
	Unlimited Multicasting		PRoPHET [8]	0.91	1387.85	2.62	0	37.69
	Limited Multicasting	Non-Social Characteristics	Spray and Wait [9]	0.80	1455.68	2.22	0	05.59
			Supernode Routing [10]	0.73	2287.84	4.29	0	38.47
		Social Characteristics	SCORP [3]	0.80	6688.25	2.34	0	00.25
ML Techniques	Limited Multicasting	Social Characteristics	SR_NB (ML technique utilized in [5])	0.93	1020.69	3.68	0	46.44
			SR_DT (ML technique utilized in [12])	0.91	1213.14	3.51	0	45.20
			SR_NN (ML Technique utilized in [12])	0.91	979.41	2.88	0	45.84
			SR_SVM	0.88	1207.29	3.28	0	45.83
			SR_RF	0.93	895.93	3.11	0	45.16
		Significant Social Characteristics	SPR_RF	**0.93**	**894.91**	**3.08**	**0**	**45.15**

are same as described in section 5.1. It is deduced that, among the different ML techniques (NB, DT, NN, SVM, RF), SR with RF (SR_RF) performs better with 93% MDP, ADD 895.93s, AHC 3.11 and OR 45.16 maintaining zero DM. It is also deduced that SPR with RF (SPR_RF) performs better compared to SR_RF and all the other existing techniques. SPR_RF compared to SR_RF shows better results with 0.1% decrease in ADD, 0.96% decrease in AHC, and 0.02% decrease in OR, while maintaining the same MDP and DM. Decrease in AHC decreases the OR, which highlights the importance of significant feature selection phase.

6. Conclusion

A socialized and efficient routing model in OMN known as Socialized Proficient Routing (SPR) is proposed in this paper. Based on the friendship between the mobile nodes, the nodes are classified into friendship and stranger nodes. Friendship nodes are selected as relay nodes for routing. Feature selection phase extracted significant features from the whole dataset and regenerated Significant Feature Dataset (SFD). The training phase trained the ML classifiers (NB, DT, NN, SVM and RF) with this SFD. The testing phase tested these classifiers with new upcoming instances that are created during routing. This is the stage where data nodes are accurately classified as friendship nodes. The performance metrics chosen for evaluating SPR are MDP, ADD, AHC, DM and OR. From the results and analysis, it is concluded that SPR_RF outperforms the other classifiers with 93% MDP, 894.91s ADD, 3.08 AHC, 0 DM and 45.15 OR. SPR_RF is also compared to SR_RF for demonstrating the significance of feature selection stage. The results concluded SPR_RF with 0.1% decrease in ADD, 0.96% decrease in AHC, and 0.02% decrease in OR, while maintaining the same MDP and DM, when compared to SR_RF. The outcome delivered is an efficient routing model in OMN known as SPR_RF using the best ML technique.

SPR works with Event Driven Simulation i.e. messages are created and routed by the ONE simulator. As future work, the performance can be evaluated by incorporating Trace Driven Simulation (TDS) in SPR. Deep learning techniques can be used rather than ML techniques in-order to train the classifiers precisely and check for the changes in performance metrics.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization, methodology, software, validation, writing, original draft preparation, Vimitha Rajendran Vidhya Lakshmi; review and editing, supervision, project administration, Gireesh Kumar Thonnuthodi.

References

[1] K. Fall, “A Delay-tolerant Network Architecture for Challenged Internets”, In: *Proc. of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pp. 27-34, 2003.

[2] V. R. V. Lakshmi and K. T. Gireesh, “Mobile Social Networks: Architecture, Privacy, Security Issues and Solutions”, *Journal of Communications*, Vol. 12, No. 9, pp. 524-531, 2017.

[3] W. Moreira, P. Mendes, and S. Sargento, “Social-aware Opportunistic Routing Protocol based on User’s Interactions and Interests”, In: *Proc. of International Conference on Ad hoc Networks*, pp.100-115, 2013.

[4] A. Socievole, E. Yoneki, F. D. Rango, and J. Crowcroft, “Ml-sor: Message Routing using Multi-layer Social Networks in Opportunistic Communications”, *Computer Networks*, Vol. 81, pp. 201-219, 2015.

[5] C. Souza, E. Mota, L. Galvao, P. Manzoni, J. C. Cano, and C. T. Calafate, “Fsf: Friendship and Selfishness Forwarding for Delay Tolerant Networks”, In: *Proc. of 2016 IEEE Symposium on Computers and Communication,* pp. 1200-1207, 2016.

[6] V. R. V. Lakshmi and K. T. Gireesh, “Opportunistic Mobile Social Networks: Architecture, Privacy, Security Issues and Future Directions”, *International Journal of Electrical and Computer Engineering,* Vol. 9, No. 2, pp. 1145-1152, 2019.

[7] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Single-copy Routing in Intermittently Connected Mobile Networks”, In: *Proc. of 2004 First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks,* pp. 235-244, 2004.

[8] A. Lindgren, A. Doria, and O. Schelen, “Probabilistic Routing in Intermittently Connected Networks”, *Service Assurance with Partial and Intermittent Resources,* pp.239-254, 2004.

[9] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Spray and Wait: An Efficient

Routing Scheme for Intermittently Connected Mobile Networks”, In: *Proc. of the 2005 ACM SIGCOMM Workshop on Delay-tolerant Networking*, pp. 252-259, 2005.

[10] D. K. Sharma, D. Kukreja, S. Chugh, and S. Kumaram, “Supernode Routing: A Grid-based Message Passing Scheme for Sparse Opportunistic Networks”, *Journal of Ambient Intelligence and Humanized Computing*, Vol. 10, No. 4, pp. 1307-1324, 2019.

[11] D. K. Sharma, S. K. Dhurandher, D. Agarwal, and K. Arora, “kROp: K-Means Clustering based Routing Protocol for Opportunistic Networks”, *Journal of Ambient Intelligence and Humanized Computing*, pp. 1-18. 2018.

[12] D. K. Sharma, S. K. Dhurandher, I. Woungang, R. K. Srivastava, A. Mohananey, and J. J. Rodrigues, “A Machine Learning-based Protocol for Efficient Routing in Opportunistic Networks”, *IEEE Systems Journal*, Vol. 12, No. 3, pp. 2207-2213, 2018.

[13] B. Mokhtar, and M. Mostafa, “Intelligence-based Routing for Smarter and Enhanced Opportunistic Network Operations”, In: *Proc. of ICSNC 2015*, pp.136, 2015.

[14] T. K. Huang, C. K. Lee, and L. J. Chen, “Prophet+: An Adaptive Prophet-based Routing Protocol for Opportunistic Network”, In: *Proc. of the 24th IEEE International Conference on Advanced Information Networking and Applications*, pp. 112-119, 2010.

[15] N. Eagle, A. S. Pentland, and D. Lazer, “Inferring Friendship Network Structure by using Mobile Phone Data”, In: *Proc. of the National Academy of Sciences*, Vol. 106, No. 36, pp. 15274-15278, 2009.

[16] M. B. Kursa and W. R. Rudnicki, “Feature Selection with the Boruta Package”, *J. Stat. Softw.*, Vol. 36, No. 11, pp. 1-13, 2010.

[17] D. Debarati, *How to Perform Feature Selection (i.e. pick important variables) using Boruta Package in R?*, 2016.

[18] J. S. Racine, “RStudio: A Platform-Independent IDE for R and Sweave”, *Journal of Applied Econometrics*, Vol. 27, No. 1, pp. 167-172, 2012.

[19] J. Han, J. Pei, and M. Kamber, *Data Mining: Concepts and Techniques*, Elsevier, 2011.

[20] A. Keränen, J. Ott, and T. Kärkkäinen, “The ONE Simulator for DTN Protocol Evaluation”, In: *Proc. of the 2nd International Conference on Simulation Tools and Techniques*, pp. 55, 2009.