A NOVEL INTERNET OF HEALTHCARE (IOH)AWARE DATA TRANSMISSION MECHANISM ININTERMITTENTLY CONNECTED WIRELESS NETWORK

Sagar Bose[1], Riya Samanta[2], Bidyut Saha[3], Chinmoy Ghorai[4] and Indrajit Banerjee[5]

[1]Department of Information Technology, Indian Institute of Engineering Science and Technology
sagar.rs2017@it.iiests.ac.in
[2]Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur
study.riya1792@gmail.com
[2]Department of Computer Science and Engineering, Calcutta University
sahabidyut999@gmail.com
[4]Department of Information Technology, Indian Institute of Engineering Science and Technology
chinmoy.ghorai@gmail.com
[5]Department of Information Technology, Indian Institute of Engineering Science and Technology
ibanerjee@it.iiests.ac.in

ABSTRACT

In multi-hop wireless networking, a hop count is a total number of intermediate nodes, especially router devices, through which data travels from source to destination. Hop count prediction is crucial for resource-constrained exigent data delivery applications like healthcare services with limited network lifetime or high processing costs. Many schemes are proposed earlier to estimate hop count for the dense or sparse networks. This paper uses Support Vector Regression (SVR) to predict hop count in an intermittently connected networking environment where end-to-end connectivity is not inevitable. A novel data forwarding mechanism is proposed for IoH aware framework. Finally, the network performance of the proposed scheme is compared with two other popular data transmission mechanisms in the same scenario, i.e., PRoPHET and PRoPHET+, for evaluation purposes. Since the SVR mechanism is computationally cheaper, shows better generalization capability, and depends only on a subset of data points (support vectors) for decision function, it is believed that this method will perform well for estimating hop count analysis and selecting minimum hop count for data delivery. The multi-feature training with dataset and predicting hop count techniques in machine learning can significantly reduce the hop count from source to destination and perform well in terms of overhead ratio and delivery probability with acceptable latency.

KEYWORDS

Hop count, Data Transmission Mechanism, Routing, Support Vector Machine, Support Vector Regression, Delay Tolerant Network (DTN), Machine Learning, intermittently connected network, internet of healthcare things.

1. INTRODUCTION

A traditional network like the internet has provided a suitable communication platform for data transmission. This provision includes routers, getaways, and wired or wireless communication-based efficient dated routing mechanisms. However, the question is whether such communication linkage and data transmission could be established where there is no continuous end-to-end path between a source node and a destination node. The answer to the question could be found in intermittently connected mobile networks, popularly known as Delay or Disruption Tolerant Networks (DTN) or opportunistic networks, which is a network of challenged networks that support long delays and data loss between and within challenged networks, has proven to be a viable alternative to infrastructure based connectivity [1]. Here an end-to-end destination provision is not guaranteed all the time that nodes are intermittently connected, or such a path is volatile and may break or change at any time, and data packets are delivered from the source node to a destination node via a store-carry-forward fashion [2]. Here, the messages or data packets are known as bundles, routed through intermediate nodes or participating nodes from a source node, enabling the bundles to store in the buffer and forward the same whenever a new participating node or a destination node comes in its communication range.

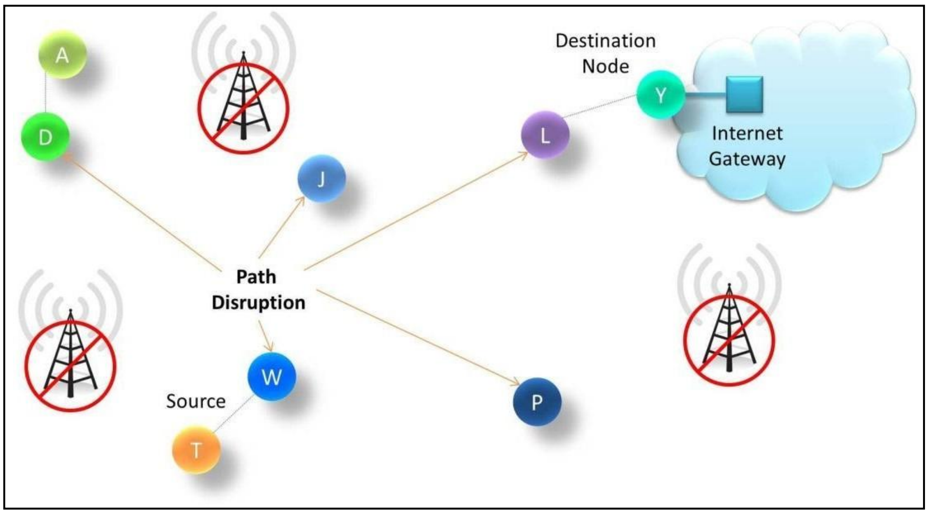

Figure 1: An example of an IoT-enabled intermittently connected networking scenario

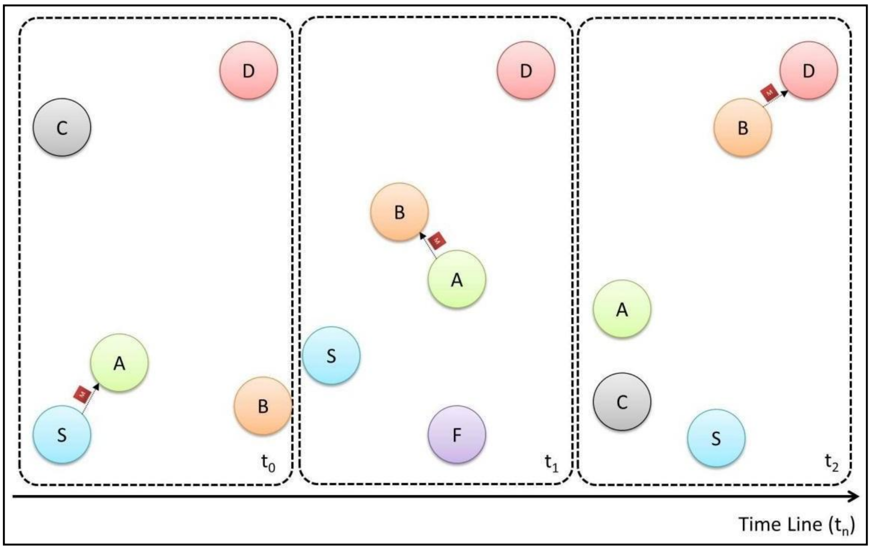

Figure 2: Illustrated example of a time-evolving store-carry-forward strategy

Figure 1 presents an Internet of Things (IoT) enabled intermittently connected networking scenario due to a crippled conventional communication backbone. There is no direct established path between a source node and the destination node, except for the partial connections between the participating nodes. Figure 2 depicts an example of a time-evolving store-carry-forward strategy based on data delivery. At time $t = t_0$ node A is in the communication range of source node S, the packet is delivered, and it is again stored and carried by A till it is interfaced with node B at $t = t_1$. Similarly, the received message is forwarded to the final destination, node D at $t = t_2$. If node D is connected with IoT Gateway, the scenario becomes ubiquitous. Examples include networks for low-cost internet provision to remote communities [3], disaster scenarios [4], interplanetary communications [5], military networks [6], wildlife tracking [7], Pocket Switched Networks [8], and vehicle ad-hoc networks (VANETs) [9], etc.

One of the most straightforward DTN dissemination protocols is the epidemic routing protocol [10]. Here a source node shares all the data packets or available bundles to all the participating nodes in its range in an epidemic fashion. The epidemic routing protocol is very effective for the participating or intermediate nodes, maintain sufficient distance among them, and has sufficient buffer memory size. However, in dense networks or participating nodes with inadequate buffer size, it shows degraded performance with significant message overhead. Further research on the same scenario shows significant Improvement over epidemic protocol, which results in suggestions with data transmission mechanisms, like Spray and Wait [11], PRoPHET protocol [12], and PRoPHET+ protocol [13]. For all these improved protocols, the store-carry-forwarding nature is maintained and limited by certain predefined conditions or predictions, and naturally, the message overhead gets better due to the limited generation of message copies. The Spray and Wait protocol classifies into two phases: *The spray* and *Wait phase*, and the total number of forwarded message copies in the network is limited. In *Spray Phase*, the source node and some other participating nodes share a limited number of message copies, which constantly receive a copy of the message. In the *Wait phase*, the participating nodes wait until the *Spray Phase* is over; that is, there is no node left that has more than one message copy, and then the message copy is delivered to a destination node only via direct transmission independently. Due to such design, this mechanism experiences prolonged delivery latency and unnecessary bandwidth consumption. The Spray and Wait routing protocol is further improved and modeled to improve the message copies spreading speed as Binary Spray and wait [14]. In Binary Spray and Wait routing protocol, only half of the message copies are disseminated from the node's buffer and shared with other participating nodes. However, in this protocol, the message copies are shared, ignoring the characteristics of receiving nodes. The PRoPHET is an acronym for probabilistic ROuting Protocol using history of encounters and Transitivity. This PRoPHET routing protocol is the first enhanced model over epidemic routing protocol by addressing minor resource demand issues and limiting message copies forwarding nature. Here the delivery probability between two participating nodes is calculated based entirely on contact history and past performances between them, where better delivery probability implies a higher probability that the two nodes come in communication range. PRoPHET allows a data packet to be delivered to another intermediate or destination node only when the delivery probability to the node is more significant than that of the source node. Such decisions help to achieve reasonable delivery probability and low overhead. However, here the dissemination speed is significantly low due to copy operation of the message depending on the satisfaction of the message delivery condition, which costs increased delivery latency and massive occupancy of buffer memory. The PRoPHET is further improved and designed APRoPHET (Advanced PRoPHET) [15], which modifies PRoPHET routing protocol parameters to stabilize its variable nature, which costs more processing time to react to changes in the scenario. PRoPHET+ is an enhanced strategy of the PRoPHET routing protocol, which was known as an adaptive probabilistic routing protocol depending on the contact history and route transitions between the participating nodes. When two participating nodes come into the communication range, the message is forwarded to the destination node if the delivery predictability value is more significant than a predefined threshold value. For multiple intermediate nodes, which are in the communication range at the same time, message copy forwards to the node with the highest delivery predictability value. As PRoPHET+ utilizes the buffer size, the power level of the nodes, location, popularity, and bandwidth availability as additional parameters to compute deliverability apart from the predictability of PRoPHET, the PRoPHET+ performs well for the delivery probability and latency, by appropriately choosing the weight factor assigned on the above parameters.

Applications of machine learning, data science, data mining, and other similar approaches are the current trend. From this aspect, we got the motivation to apply machine learning-based algorithms to research thoroughly for estimating the best hop-count towards the final destination in the substantial amount of network and routing data in an intermittently connected network. For an accurate hop count prediction model, complex mapping relationships between hop counts and data routing features are identified through machine learning-based training and predicting techniques to identify a precise and best route towards the desired IoT-enabled gateway. In this paper, the SVR machine learning model is applied with Radial Basis Function (RBF) kernel to determine the relationship between the hop count parameter and values of some data routing features for efficient data delivery in an intermittently connected network. Compared with other machine learning models, the kernel is done in selecting SVR with the RBF as the relationship between the hop count parameter and other network features is non-linear. The hop count target value prediction using SVR is implemented by tuning its hyper-parameters, mainly gamma, C, and epsilon. The data set for training and testing is prepared by executing SVR on the collected data set by simulating the Epidemic Routing protocol. The entire data set were divided randomly with a 4:1 ratio to generate training data set used to train the model, and a testing data set used to assess the hop count prediction. After estimating hop count, a novel data transmission mechanism was proposed using the estimated hop count value in an IoH-aware framework, and finally, it was evaluated.

In addition to this introduction, the remainder of the paper is organized as follows. Section 2 introduces related studies. Section 3 proposes an architecture for IOH-aware intermittently connected networking scenarios. Section 4 depicts the proposed data transmission mechanism by predicting hop count using a machine learning algorithm with its mathematical model. Performance evaluations are discussed, and simulation results are shown in section 4. Finally, Section 5 concludes our paper.

A. Contribution

Thus, in summary, the contributions of the work reported in this paper are as follows:

1) It employs SVR machine learning for estimating hop count in intermittently connected wireless networks to support the internet of healthcare aware architecture.

2) It formally defines the minimum hop-count-based data transmission mechanism toward the IoT-enabled destination and thereby fills the gap between the perception layer and the application layer to reach the pervasive nature of physiological or healthcare data accessibility.

3) The method's performance is evaluated by simulating the proposed mechanism, which outperforms the other two similar data routing schemes, PRoPHET and PRoPHET+ place, and shows a relatively high delivery ratio and low overhead with acceptable average end-to-end latency.

1. RELATED WORKS

Two popular routing protocols have been cautiously chosen for our comparisons and evaluation that closely meet the rural health support or post-disaster situation framework supporting an intermittently connected networking environment. Such protocols are PRoPHET and PRoPHET+. Moreover, another routing protocol, Epidemic, for the same scenario, is simulated to generate a training dataset for machine learning. These are discussed in Sections 2.1-2.3. In section 2.4, we discuss the existing works related to the data routing schemes by estimating hop count.

Epidemic Routing Protocol

The Epidemic Routing (ER) technique is one of the earliest strategies of DTN routing protocol, which forwards or delivers a message by flooding techniques across the network. Here the basic idea is that it spreads messages like the transmission of a disease, where it forwards message copies to every encountered neighboring node. A source node assigns a maximum number of hop counts, which a message copy can travel to reach the sink node. This multi-hopping or traveling of the message copy is repeated until the message copy is finally delivered to the destination or maximum hop-count is achieved, that is, till the Time-to-live (TTL), after which the message copy will be deleted. Each message is stored in the buffer memory with an exclusive identifier known as the Summary Vector. Before forwarding a message, Indices of all buffer messages are exchanged between the two participating nodes to check the availability of the Vector, and thus the sharing of the message is deleted.

The Infocast protocol [16] improves the epidemic protocol with coding and fragmentation. It provides a lower bound for the delay and an upper bound for delivery probability with a cost of wastage of resources like buffer size, battery energy, etc., and high overhead cost with traffic congestion as the message copies are forwarded to all the participating nodes. Several enhancement is done to the ER to regulate the wastage issue of the resources. Four different approaches are proposed to restrict copy and forwarding of the already delivered messages in [17]. In the same way in [18], it is proposed that when the probability is less than 1, message copies will be shared. Priority based message copy forwarding strategy is employed in POR [19], MaxProp [20], DAER [21] and RAPID [22].

PRoPHET Protocol

PRoPHET, a probabilistic routing protocol, is the first context-aware protocol that uses contact history to calculate the delivery predictability $P \in [0, 1]$ which denotes how likely a participating node will be able to deliver a message to a destination, which is updated according to equation 1. If a user follows a path or visits a node that follows a particular movement pattern, it can be predicted that it will follow that again. Hence, the routing performance can be improved by knowing the contact history of the movement pattern. At every source node α, when a message is forwarded to a destination node β, the delivery predictability:

$$P(\alpha,\beta) = \begin{cases} P(\alpha,\beta)_{Old} + (1 - P(\alpha,\beta)_{Old})P_{init}, & if\ \alpha,\beta\ meets \\ \gamma^k P(\alpha,\beta)_{Old}, & Otherwise \end{cases} \quad (1)$$

Where P_{init} is the initial predictability γ is the aging factor ($\gamma < 1$) and k is the time since the last update. If a source node X interacts with node Y for sending its message to the final destination Z, node X, and node Y initially exchange summary vectors and delivery predictability. Then node X Compares $P_{X,Z}$, and $P_{Y,Z}$. $P_{Y,Z} > P_{X,Z}$, then the message copy is forwarded to node Y. Otherwise it is not copied. It incurs huge buffer occupancy due to a lack of acknowledgment. The PRoPHET is further enhanced by incorporating the history of messages [23] to improve the delivery chance. The message forwarding decision is taken based on the message's hop-count history and delivery predictability. Introducing a utility factor to the PRoPHET protocol for optimization and prediction of message selection by employing message scheduling and message dropping mechanism is done in the paper [24]. It performs well for buffer management.

PRoPHET+ Protocol

PRoPHET+, an improved version of PRoPHET routing protocol predicts deliverability based on buffer size (V_B) battery power level (V_P) popularity (V_O), bandwidth (V_A), and delivery predictability of PRoPHET ($V_{PRoPHET}$). Using additive weighting and the utility function is as follows:

$$V_d = W_B(V_B) + W_P(V_P) + W_O(V_O) + W_A(V_A) + W_{PRoPHET}(V_{PRoPHET}) \quad (2)$$

Where W_i refers to the weightage which depends on the scenario and is configured by the user. While interacting, a source node enquires deliverability value and forwards the message only when it exceeds a predefined threshold value. For multiple nodes, the message is forwarded to the node with the highest deliverability value. It performs well from the aspect of delivery probability value and average delay but strictly depends on the selection of the weightage parameters.

Hop-count-based data transmission mechanism

Apart from the study in intermittently connected networking framework, estimating hop count is an important design criterion in multi-hop wireless networks. Other application-oriented research is done based on hop count parameters targeting different types of wireless networks. Like, for infrastructure-based Wireless Mesh Network (WMN), a hop count-based congestion aware strategy is designed to discover the path to the internet Gateway in [7]. Similarly, for infrastructure-less primary ad-hoc networks, a hop count-based investigation is done for position-based routing to maximize delivery probability [1]. Moreover, for Mobile Ad-hoc NETwork (MANET), distance and direction-based routing strategies are used to select the next hop [3]. Here hop count is considered a significant contributing factor to the position-based routing protocol. Some enhancement of the existing routing protocol is also investigated based on the hop count parameter [4] in the same scenario. However, the target application area is mainly city based dense network. Again some further research is done in mobile DTN for making heuristic estimation based on the hop count information [2] and sparse network to define hop count-based path selection.

***Synthesis*:** We synthesize that there exists a research lacuna in statistical learning-based hop count estimation and efficient data transmission mechanism in an intermittently connected networking environment. Therefore, though different schemes are proposed for modeling hop count and thereby determining the best possible path for data delivery with better network performance, none of the mechanisms are satisfying in continuous learning based on improved and practicable models. Thus, a store-carry-forward-based data transmission mechanism in IoH aware framework is proposed in this paper.

2. IoH aware Intermittently Connected Networking Architecture

The objective of IoH-aware intermittently connected network infrastructure and the communication link is to support competent healthcare services during a post-disaster rural health scenario where conventional communication is partially paralyzed or broken. Three-layer architecture has been envisioned under the current research work over which an efficient routing algorithm for the intermittently connected network has been implemented. Figure 3 depicts the architecture having three functional layers, which are the perception layer, intermittently connected network layer, and remote access IoH application layer. The communication strategy for each layer is discussed as follows:

Communication framework in perception layer: The lowest layer of the infrastructure is the perception layer, where physiological sensor data packets are generated from wireless-enabled wearable physical sensor nodes, carried by the paramedic staff to support primary medical management services to disaster-affected or remote patients. In particular, we envision patients wearing the node, capable of generating vital physiological sensor information like body temperature, blood pressure, SPO2, ECG, heartbeats, movement or motion, etc., in the form of calibrated sensor data packets or messages (M).

Communication framework in intermittently connected network layer: Ms' are then efficiently traveled in the intermittently connected network layer following store- carry-forward fashion. The destination of the bundles is either the cloud-enabled access points or medical kiosks to interface with the application layer. Here are the access points on medical kiosks acting as IoT intelligent gateways. Ambulance services this year may also act as an intermediate mobile sink to expedite the delivery date.

Communication framework in remote access IoH application layer: To enable cloud-assisted intelligent healthcare services, the data packets are finally stored, filtered, processed, and analyzed at the distributed cloud-assisted machines or medical control station engines. Healthcare giving assistance is generated from a full-featured hospital in the same layer. Thus IoH assisted medical management services are remotely and efficiently organized and managed.

Thus the network scenario discussed for rural or post-disaster health caregiving architecture can build a communication process among the distributed and intermittently connected wireless nodes to finally interface with the cloud so that the Internet of Things is enabled. It may be noted that some part of the proposed architecture design has been filed as a patent [34].

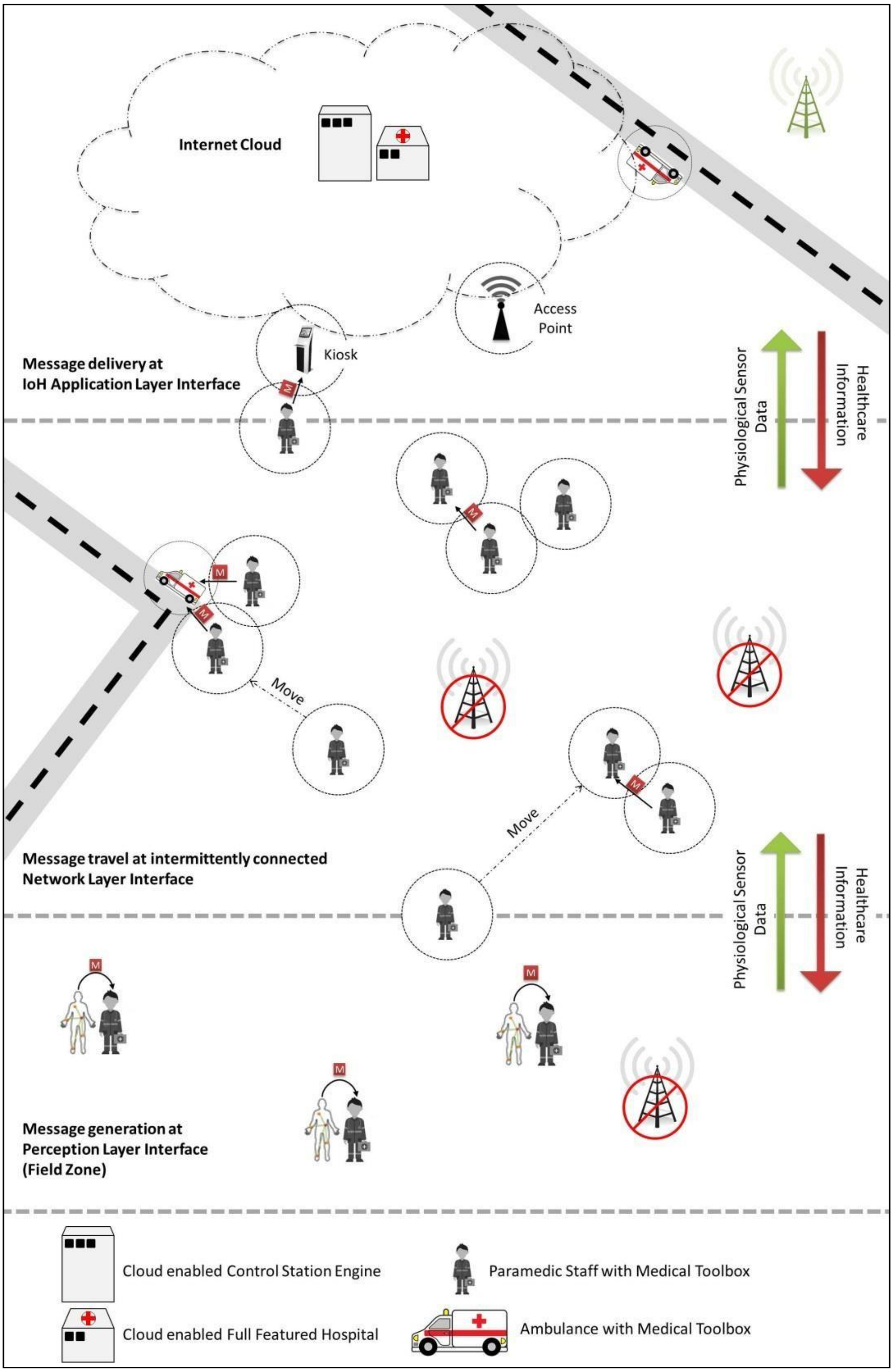

Figure 3: Architecture for IoH aware intermittently connected networking scenario

3. PROPOSED DATA TRANSMISSION MECHANISM BY PREDICTING HOP COUNT USING SVR

For an efficient data transmission mechanism, the sender node selects the forwarder node by effectively predicting the hop count for its neighboring nodes to reach the desired destination. Then the best-selected candidate node covering minimum hop count forwards data, while the rest of the neighboring nodes are discarded. The framework of the proposed data transmission mechanism is shown in figure 4, which consists of three modules: fault-tolerant forwarder node qualifier module, hop count predictor model using SVR algorithm, and forwarder node selector module in IoH aware intermittently connected wireless networking scenario. In the following sections, they are subsequently discussed.

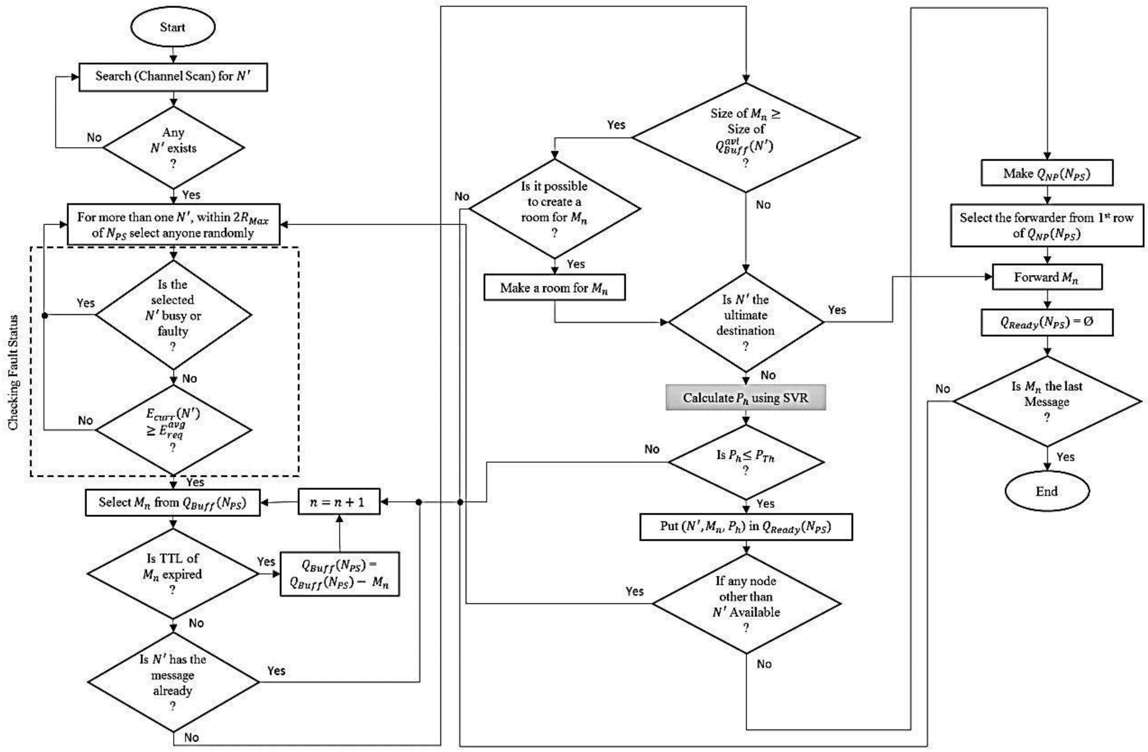

Figure 4: Flowchart of the message forwarding mechanism based on predicted hop count using SVR

Fault tolerant forwarder node qualifier module

When a sender wireless node carried by paramedic stuff (N_{PS}) wants to send a message to a certain destination, it starts scanning its wireless channels for a possible forwarder node and this searching continues until it finds any possible forwarder node (N') which may be carried by paramedic stuff or an ambulance. If it finds more than one N', then it selects any one node randomly and at first, it performs testing for fault status. This is done by checking whether N' is busy or faulty in terms of hardware or software fault or having sufficient residual energy ($E_{curr}(N')$ or not. The required average energy (E_{req}^{avg}) depends on the transmission energy, node density level in the network, message size etc. If the selected possible forwarder node founds healthy, then N_{PS} selects a message (M_n) from its buffer queue $(Q_{Buff}(N_{PS}))$ and checks whether the time-to-live (TTL) of the message is expired or not. If it is already expired then N_{PS} removes M_n from $Q_{Buff}(N_{PS})$ and selects the next message, otherwise sends a message summary to N' which contains message ID, source node address and destination node address information. After receiving the message summary information, the receiver checks whether the same message already resides in the buffer memory or not. If it matches, then asks N_{PS} node to select the next message, otherwise, it checks the size of the message. If N' finds the size of M_n is more than the available buffer size $(Q^{avl}_{Buff}\ (N'))$ then again analyze whether there is any possibility to create the required memory space or room by dropping some old message(s) or not. If it is not possible then ask N_{PS} to select next message. In case, if it finds available room for M_n, review whether the destination node information of M_n matches its own address or not. If it is the same then N_{PS} forwards M_n and the next message is selected. If not N' qualifies the criteria required for considering it as a possible healthy forwarder node.

Hop count predictor module using SVR algorithm

Before estimating the predictable hop count value (P_h), a maximum threshold value of hop count (h_{Th}) is set which may require travel towards the destination node. h_{Th} is decided based on the application scenario, number of nodes, pattern of nodes, range of nodes, and the size of the area. For possible healthy forwarder N', the N_{PS} calculates a predictable hop count value using the SVR algorithm to reach the desired destination node. For estimating P_h data collection and feature selection are done by selecting required network statistics as the input for the SVR-based predictor module and finally the predictor module predicts the required hop count value as the input for the forwarder node selector module. The design of each module is described in the following sections.

(i) Data collection phase

The parameter collected in this phase is based on the intermittently connected networking scenario. So, to design the SVR-based machine learning mechanism, the raw data collection is carried out by simulating epidemic routing protocol with a cluster-based mobility model. The epidemic routing protocol is chosen to take advantage of flooding property; that is, the message is flooded in every possible way; so gathering data for learning is one of the best approaches. Furthermore, the cluster-based mobility model is followed as the tendency of human nature is to move in a group due to social status, profession, relationship, etc. The cluster mobility model outperforms another available mobility model in the context of human mobility in an emergency scenario. After the simulation, all the text-log files are collected for processing. Selective data from the results merged to make a master table which is again randomly divided with a ratio of 4:1 to generate a training and testing dataset to train the SVR model and assess the predicted hop count result, respectively. The predicted hop count values are verified with more than 90% accuracy. The framework for the data collection and dataset generation is depicted in figure 5.

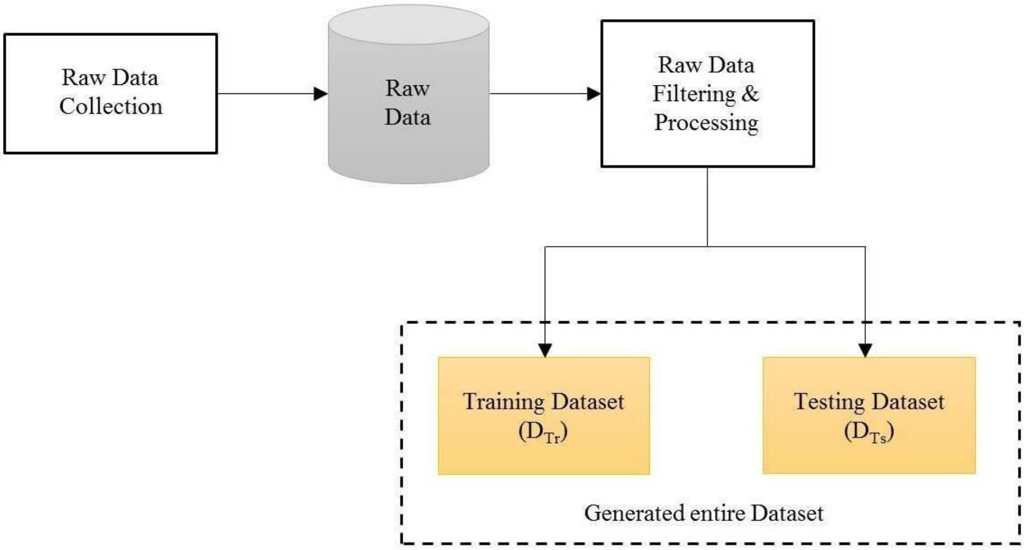

Figure 5: The framework for data collection and dataset generation

(ii) Feature selection phase

The final selected features of the collected data will be used as the input during the development of the SVR algorithm-based hop count prediction phase, which is listed in table 1 and described with a layout in figure 6. Here a source node X generates a message M with features like message TTL M_T and message size M_B. As nodes Y, Z and L are within the communication range of node X, Y, Z, and L are the probable forwarder node.

Table 1: Features of the collected data

Network Parameters	**Selected feature identity**	**Feature explanation**
Message	M_T	Message TTL
	M_B	Message size
Forwarder Node	x_{f1}	x coordinate of the forwarder node source location
	y_{f1}	y coordinate of the forwarder node source location
	x_{f2}	x coordinate of the forwarder node destination location
	y_{f2}	y coordinate of the forwarder node destination location
	F_B	Buffer Occupancy
	F_t	Next time to move (for forwarder node)
Destination Node	x_{d1}	x coordinate of the destination node source location
	y_{d1}	y coordinate of the destination node source location
	x_{d2}	x coordinate of the destination node destination location
	y_{d2}	y coordinate of the destination node destination location
	D_t	Next time to move (for destination node)

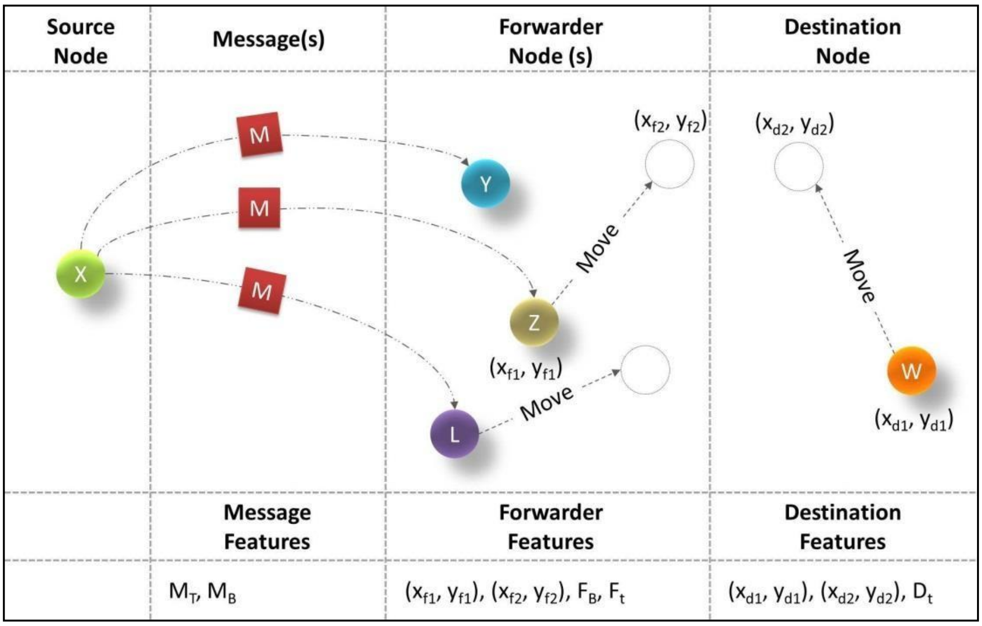

Figure 6: Feature description with an example in intermittently connected networking scenario

(iii) SVR algorithm based hop count prediction phase

The core of the data transmission mechanism, which significantly impacts the system's overall performance, is the hop count prediction phase using the SVR algorithm. A Support Victor machine (SVM), initially developed to solve classification problems, is a supervisor machine learning model derived from the statistical learning theory and statistical risk minimization principle [25], which is later widely used to solve regression and abnormality detection problems. SVM can be classified as linear SVMs' to address linearly separable data objects with the linear classifier, which has the largest distance in feature space and non-linear SVMs' to address the non-linear classification problems by transforming linearly non-separable problems in low dimensional space into linearly separable problems in high dimensional space using a kernel function. SVR is a particular case of SVM, which is developed to solve regression problems. The regression method determines the relationship between the system input and output from a reliable training dataset. The predicted system output should be as close as possible to the practical value [26]. To predict the hop count value, the non-linear SVR algorithm is used due to the nature of complex characteristics of network parameters in an intermittently connected wireless network. Linear regression is developed from a non-linear regression by implementing kernel function. Then, to meet the predicted space, a hyperplane is used, and for consecutive prediction, the tuned hyper-parameters are applied [28]. The use of SVR-based hop count prediction with the kernel function selection and hyper-parameter tuning steps are described in detail in the following sub-sections.

(a) Description of SVR:

The basic idea of SVR is to implement non-linear class boundaries by adopting a linear model and the map input vector set x_{in} into a high dimensional feature space, where a non-linear model is transformed into a linear model. After that a quadratic programming problem is developed from the regression problem, which is easily solvable. In this paper the training vector from input space $x_{in} = [(M_B, M_T), \{(x_{f1}, y_{f1}), (x_{f2}, y_{f2}), F_B, F_t\}, \{(x_{d1}, y_{d1}), (x_{d2}, y_{d2}), D_t\}]$ is mapped into a high dimensional feature space and then form a separating hyperplane as shown in figure 7.

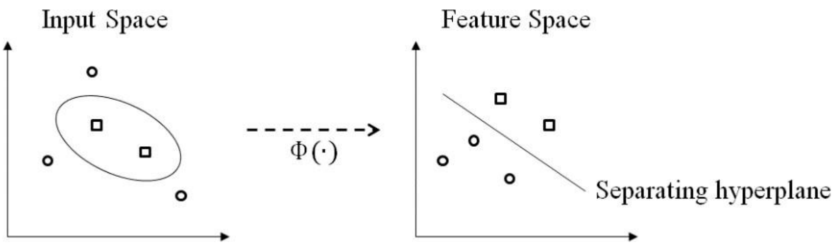

Figure 7. The fundamental concept of SVR (classify circular and square structure)

Given training dataset

$$D_{Tr} = (x_{in}, C_h)|^{l}_{i=1}, \qquad i = 1, 2, \dots, l;\ x_{in} \in R^l, y_{out} \in R \tag{1}$$

Here, C_h is the hop count value, l = total number of samples in the training data set.

The goal of the function regression is to find the mapping $f: R^l \rightarrow R$ and make $f(x_{in}) \approx C_h$. This mapping relationship is non-linear in nature. The SVR maps x into higher dimensional feature space (F) by utilizing a non-linear mapping function $\phi(x)$ and then the linear regression is performed in the feature space.

$$P_h = \mathrm{f}(x) =< \omega^{\mathrm{T}}.\phi(x) > + \mathrm{b} \qquad \phi: \mathrm{R}^{\mathrm{n}} \rightarrow \mathrm{F}, \omega \in \mathrm{F} \tag{2}$$

Where, ω^T and b are the weighting matrix and bias of the support vector, respectively. $<.,.>$ = Dot product operation in F

The objective of SVR is to determine a suitable ω and b such that P_h is as near as possible to C_h. Parameter ω and b are calculated by minimizing the following regularized risk function:

$$R(\omega) = \frac{1}{2}\omega^T\omega + \lambda \sum_{i=1}^{l} |C_h - (\omega^{\mathrm{T}}.\phi(x) + b)|_s \tag{3}$$

Where, λ and ε are denoting emperical parameters and $|C_h - (\omega^{\mathrm{T}}.\phi(x) + b)|_s$ means the insensitive loss function [35].

$$|C_h - (\omega^{\mathrm{T}}.\phi(x) + b)|_s = \begin{cases} 0 & |C_h-(\omega^{\mathrm{T}}.\phi(x)+b)|<s \\ |C_h - (\omega^{\mathrm{T}}.\phi(x) + b)| - \varepsilon, & |P_h-(\omega^{\mathrm{T}}.\phi(x)+b)\geq s \end{cases} \tag{4}$$

When the predictive error is less than ε; the value of the loss function becomes 0; otherwise linear punishment is applied. For minimizing equation (4), two variables are introduced, ξ_i^+, ξ_i^-, which are also known as slack variables, as depicted in figure 8. So, equation (4) can be deduced as

$$R(\omega) = \frac{1}{2}\omega^T\omega + C\sum_{i=1}(\xi_i^+ + \xi_i^-) \quad (5)$$

Subject to

$$\begin{cases} C_h - \omega^{\mathrm{T}}.\phi(x_i) \leq \varepsilon + \xi_i^+ \\ \omega^{\mathrm{T}}.\phi(x_i) - C_h \leq \varepsilon + \xi_i^- \end{cases} \quad \xi_i^+ \geq 0, \xi_i^- \geq 0, i = 1, \dots, l. \quad (6)$$

Where C is regularization constant.

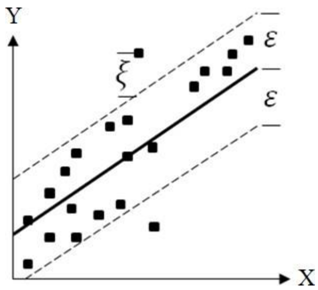

Figure 8. Slack variable for soft-margin SVR

By introducing Lagrangian function, the above optimization problem can be defined as:

$$\begin{aligned} L(\omega, \xi_i^+, \xi_i^-, \alpha^+, \alpha^-, r^+, r^-) &= \frac{1}{2}\omega^T\omega + C\left(\sum_{i=1}^{l}(\xi_i^+ + \xi_i^-)\right) - \left(\sum_{i=1}^{l}(r^+\xi_i^+ + r^-\xi_i^-)\right) \\ &- \sum_{i=1}^{l}\alpha_i^+(\omega^{\mathrm{T}}.\phi(x_i) - C_h + (\varepsilon + \xi_i^+)) \\ &- \sum_{i=1}^{l}\alpha_i^-(C_h - \omega^{\mathrm{T}}.\phi(x_i) + \varepsilon + \xi_i^-) \end{aligned} \quad (7)$$

Here, above optimization must satisfy:

$$\begin{aligned} \frac{\partial L_M}{\partial \omega} &= \omega - \sum_{i=1}(\alpha_i^+ - \alpha_i^-)\phi(x_i) = 0 \\ \frac{\partial L_M}{\partial \xi_i^+} &= r^+ - (C - \alpha_i^+) = 0 \\ \frac{\partial L_M}{\partial \xi_i^-} &= r^- - (C - \alpha_i^-) = 0 \end{aligned} \quad (8)$$

Where, $L_M = L(\omega, \xi_i^+, \xi_i^-, \alpha^+, \alpha^-, r, r')$, and $\alpha^+, \alpha^-, r^+, r^-$are Lagrange multiplier.

By substituting (8) into (7) the problem can be transformed to dual optimization problem. A convex function can be obtained:

$$Q(\alpha^{+}, \alpha^{-}) = \sum_{i=1} C_h (\alpha_i^{+} - \alpha_i^{-}) - \varepsilon \sum_{i=1} (\alpha_i^{+} + \alpha_i^{-}) - \frac{1}{2} \sum_{i=1} \sum_{j=1}^{l} \{(\alpha_i^{+} - \alpha_i^{-})(\alpha_j^{+} - \alpha_j^{-}) K(x_i, x_j) \quad (9)$$

Subject to

$$\sum_{i=1}^{l} (\alpha_i^{+} - \alpha_i^{-}) = 0, \qquad 0 \le \alpha_i^{+} \le C, 0 \le \alpha_i^{-} \le C, i = 1,2,, \ldots\ldots, l \qquad (10)$$

Hence, the estimated target output can be represented by:

$$P_h = \sum_{i=1}^{l} (\alpha_i^{+} - \alpha_i^{-}) K(x, x_i) + b \qquad (11)$$

Where $K(x, x_i)$ is the kernel function.

(b) Kernel function selection:

Kernel function is used to map the initial data in a feature space into a higher dimensional space where data are considered as lineal separable. Generally three different types of kernel functions are considered for the SVR algorithm which is listed in table 2 with specific details.

Table 2: Three different kernels

Kernel Functions	**Formulation**	**Parameter**
Polynomial Function	$K(x_i, x_j) = (\gamma(x_i, x_j) + l)^d$	γ, l, d
Radial Basis Function (RBF)	$K(x_i, x_j) = \exp\ (-\gamma \vert x_i - x_j\vert^2)$	γ
Sigmoidal Function	$K(x_i, x_j) = \tanh(\gamma(x_i, x_j) + l)$	γ, l

To predict hop count using is SVR algorithm by mapping the original feature space $x_{in} = [(M_B, M_T), \{(x_{f1}, y_{f1}), (x_{f2}, y_{f2}), F_B, F_t\}, \{(x_{d1}, y_{d1}), (x_{d2}, y_{d2}), D_t\}]$ onto the new feature space $x_{in} = (x_1, x_2, x_3, \ldots x_l)$, the RBF kernel function is adopted. To establish a non-linear mapping model between network parameters and hop count value, the finite set of hop count value can be expressed by a linear regression formula in the new feature space. As shown in table 2, the RBF kernel function has the advantage of fewer parameters than that of the Polynomial and Sigmoid kernel functions, which eventually reduces the complexity of model selection. Therefore, the RBF kernel function has the lowest complexity of the other two kernel functions.

(c) Hyper-parameter tuning:

For the implementation of SVR, RBF function based three parameters are important, gamma, C, and epsilon. The gamma parameter is used for similarity measure which defines whether the single training example is far or close. As gamma is set as auto so it uses 0.08. The small value of gamma implies the significant influence of the training samples. Another important parameter for all kernel function is C which controls or regulates the error due to training and testing. Low value of C allows more outliers, whereas large value of C allows fewer outliers. C value uses a 13.0 which makes smoother decision surface. Epsilon parameter indicates maximum allowable error value per training data instance and set as 0.10.

(d) Hop count prediction process:

The detailed step-by-step hop count prediction process is depicted in figure 9, which includes 4 steps:

Step 1: After gathering raw data, filtering and processing is done and then features are extracted. Then it is divided into a training dataset and testing dataset with 4:1 ratio.

Step 2: To establish SVR model for hop count prediction, RBF kernel function is selected and C, gamma and epsilon hyper-parameters are tuned.

Step 3: The hop count prediction model is obtained from the training process by feeding the training dataset with the selected features and hop count target vector.

Step 4: Then testing dataset is applied to validate the predicted hop count output from the prediction model.

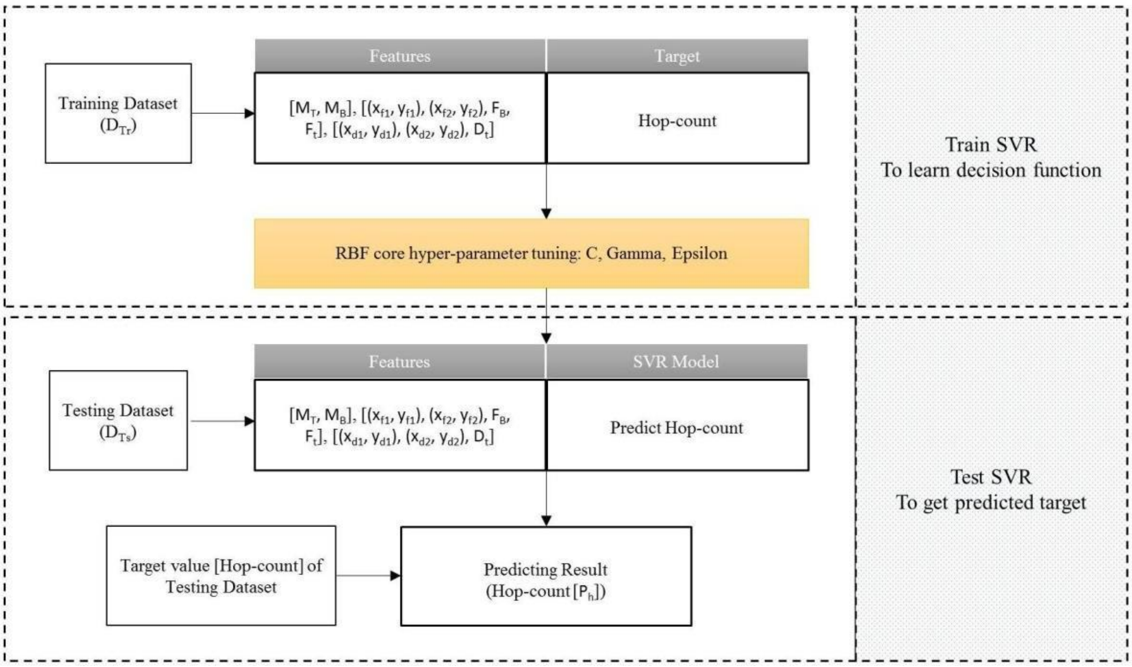

Figure 9. Hop count prediction using SVR

Forwarder node selector module

To select the best forwarder node, at first P_h is compared with the h_{Th} value and if $P_h < h_{Th}$, then it is tabulated in a ready queue ($Q_{Ready}(N_{PS})$) with the M_n, otherwise the message is not forwarded and next message is selected. by doing this, more forwarding restrictions are introduced in the proposed protocol then other traditional similar protocols like PRoPHET, PRoPHET+ etc. and thus excessive message forwarding is avoided. Then N_{PS} scans for availability of other wireless nodes within its communication range. If it finds any node other than N', follows the same steps as for N' and like these the $(Q_{Ready}(N_{PS}))$ is made until it exhausts for all the nodes available in its range. From this, $Q_{Ready}(N_{PS})$), a node preference queue $(Q_{NP}(N_{PS}))$ is formed to select the best forwarder node and thereby the best route with the minimum hop count and M_n is forwarded. After this step, thread queue is deleted. If M_n is not the last message, the same steps are followed for the remaining messages in the buffer queue.

Figure 10 shows example of forming the ready queue. Here, node A is connected with nodes B, C, D, and E with the maximum range covering $2R_{max}$. A message M_{AT} is generated with destination node T. Here B, C, D, and E are the possible forwarder node. The ready queue of A is formed with the predicted hop count through all the forwarded nodes. Then the queue is sorted based on ascending order of P_h values and forms in the node preference queue. The forwarder node B from the first row of the node preference queue is selected and M_{AT} is forwarded. Different types of node connections and their corresponding layout are shown in figure 11.

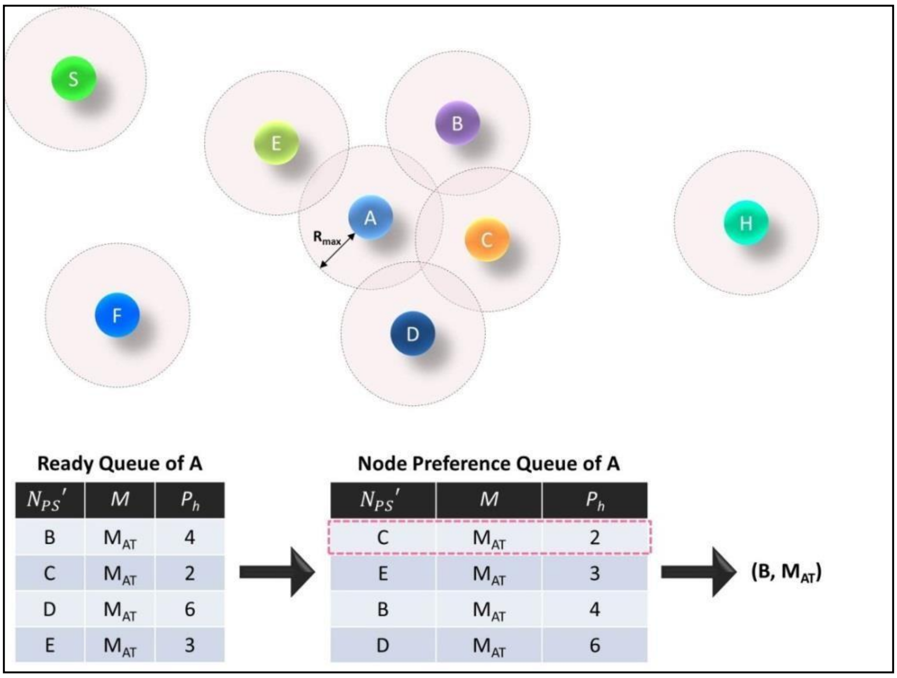

Figure 10. Example of a forwarding scenario

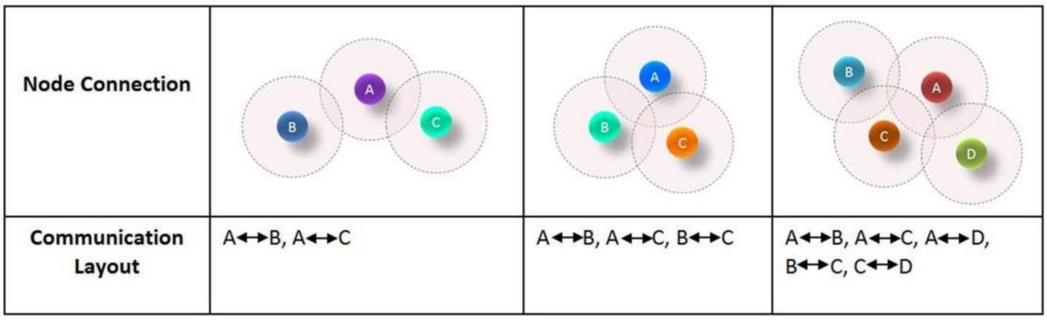

Figure 11. Node connection and communication layout

4. PERFORMANCE EVALUATION

5.1. Simulation environment set up

In this section the performance of the proposed data transmission mechanism is analyzed and then compared with performance with the PRoPHET and PRoPHET+ routing protocol, which are contact history information based latest and popular protocols in intermittently connected networking scenario. For the implementation of proposed SVR machine learning based data transmission mechanism, a cross platform environment was adopted, which was implemented by calling REST service to bind Python language based machine learning for hop count prediction and Java language based software to simulate the data transmission mechanism in intermittently connected network. For the implementation of machine learning, SciKit – Learn Library package [31] in python was adopted and then for implementing the proposed data transmission mechanism as well as evaluating its performance by implementing the PRoPHET and PRoPHET+, protocol Opportunistic Network Environment (ONE) simulator [29, 30], a Java-based software developed by Helsinki University, was used, which is one of the most preferred simulation software for DTN. The training and testing dataset were also generated by simulating Epidemic Routing protocol using ONE simulator software. The parameter values assumed for the simulation are listed in table 3.

Table 1: The Simulation parameter values.

Parameter	Value
Area size (m^2)	4500 X 3400
Movement model	Cluster based model
Speed (m/s)	Nodes in static groups: [0.5, 1.5] carrier nodes: [0 – 15, 15 – 30, 30 – 45, 60 – 75]
Number of nodes	In each static group: 25 In carriers: 10
Simulation time (s)	3000

Packet transmission speed (Kbyte/s)	250
Message generation rate or interval (s)	25, 120
Event generator class	Message Event Generator
Message size (Byte)	50k ~ 1M
Buffer size (Byte)	(50, 100, 150, 200, 500)M
Message TTL	240min
Number of interfaces	1 for each group
Number of host groups	5 (4 static groups and 1 carrier)

For ONE simulator, the entire area is 4500 x 3400 m^2 (default) which includes four clusters. Four groups and created for simulation purpose: Fields Zone (for disaster scenario it is disaster affected area or for rural health support framework, it is the area where healthcare is needed), Primary Health Care Center, full featured hospital and Control Station Center. It is assumed that the carrier like paramedic staffs or ambulances are moving within each clusters with a wireless node and so forth inter-cluster communication is maintained. The algorithm runs on a computer with a Linux (Ubuntu) operating system, 4GB RAM, INTEL® CORE^{TM} i-5 CPU.

Simulation Results

Due to Mobile in nature, the resources of participating nodes such as buffer size, power available etc is limited. Therefore, different parameters need to be tuned for different applications for designing intermittently connected network scenario. For some cases, Network overhead is important factor with allowable delivery probability value limit, some others preferred for increasing delivery probability with considerable latency average etc. The performance of the proposed data transmission method is compared with the PRoPHET and PRoPHET+ protocol in terms of hop-count average, overhead ratio, latency average and delivery ratio metrics by varying buffer size, transmission range and transmission speed. The hop count value, overhead ratio, delivery latency and delivery ratio can be defined as follows:

$$\text{Delivery ratio} = \frac{\text{Total number of successfully delivered data packets}}{\text{Total number of data packets generated}}$$

$$\text{Overhead Ratio} = \frac{\left(\begin{array}{c}\text{Total number of the}\\ \text{data packets forwarded}\end{array}\right) - \left(\begin{array}{c}\text{Total number of the data}\\ \text{packets successfully delivered}\end{array}\right)}{\text{Total number of data packets successfully delivered}}$$

$$\text{Delivery latency} = \frac{\text{Sum of delays for successfully delivered data packets}}{\text{Total Number of successfully delivered data packets}}$$

$$\text{Hop count value} = \left(\begin{array}{c}\text{Total number of nodes for forming} \\ \text{a successful data routing path}\end{array}\right) - 1$$

Based on the network parameters described in table 1, the simulations have been performed for three data transmission mechanisms and the mean results are graphically presented in figure 12, by varying buffer size, transmission range and transmission speed.

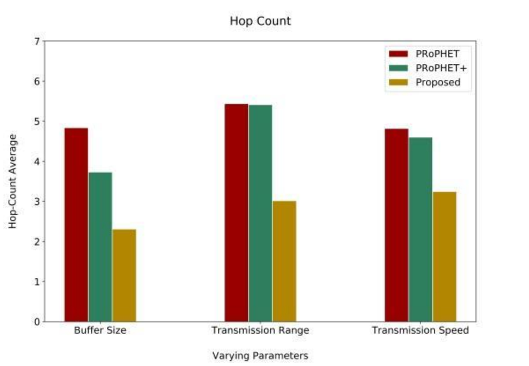

Figure 12(a). Evaluation for mean hop count

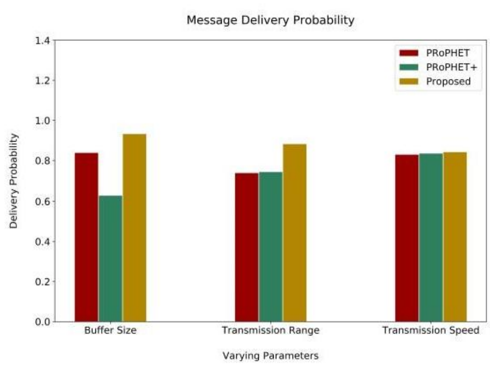

Figure 12(a). Evaluation for mean message delivery probability

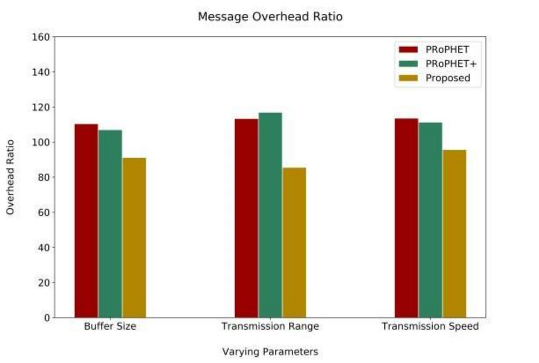

Figure 12(a). Evaluation for mean message overhead ratio

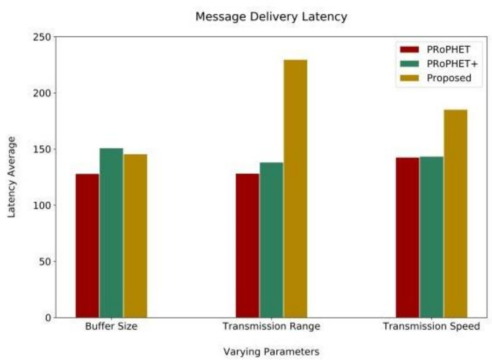

Figure 12(a). Evaluation for mean message delivery latency

In this paperwork, by means of controlling the hop count, we try to reduce overhead while maintaining acceptable average delay with satisfactory delivery ratio (delivery ratio > 80%). To effectively evaluate, each variations are analyzed in detail in the following sections.

Evaluation with different Buffer Size

Figure 5 demonstrates the performances by varying the buffer size from 50MB to 500MB. It is evident that no matter how the buffer size changes, the proposed scheme maintains lowest level in hop-count average due to it's effective machine learning based data routing strategy, compared with PRoPHET and PRoPHET+, as shown in figure 5(a). While increasing buffer size the tendency of message drop decreases as well as delivery probability increases as evident in figure 5(d), therefore naturally the overhead ratio decreases which is reflected in the outcome shown in figure 5(b). But due to machine learning based constraint data dissemination mechanism, proposed scheme achieves higher latency average than PRoPHET and outperforms PRoPHET+, as shown in figure 5(c).

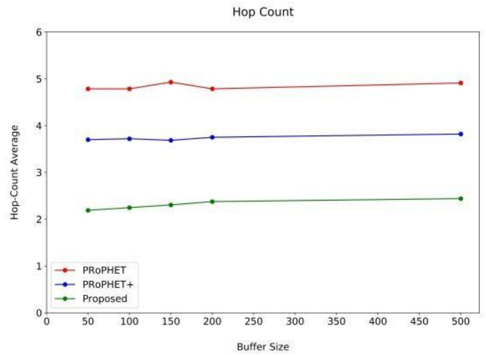

Figure 5(a): Buffer Size versus Delivery Ratio

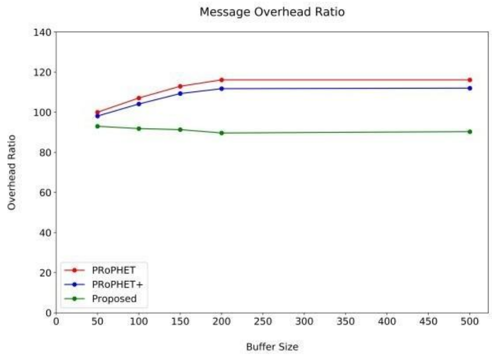

Figure 5(b): Buffer Size versus Overhead Ratio

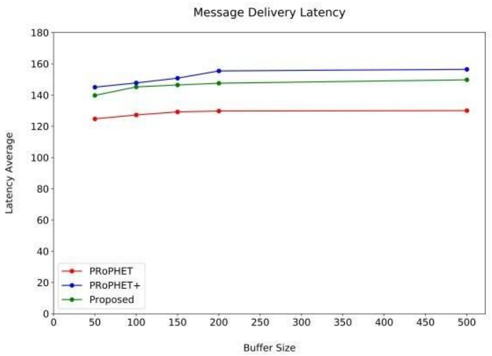

Figure 5(c): Buffer Size versus Latency Average

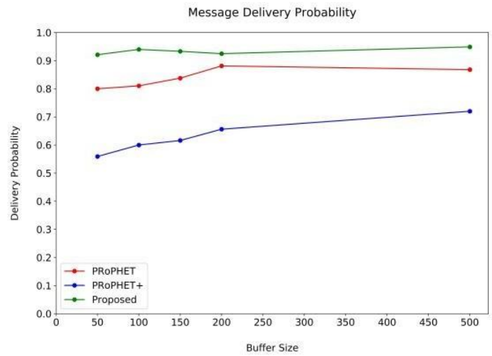

Figure 5(d): Buffer Size versus Hop Count Average

Evaluation with different Transmission Range

As presented in figure 6, by varying transmission range, naturally node coverage area increases and consequently average latency decreases as shown in figure 6(c), and as a result, delivery ratio increases and overhead ratio decreases, which is depicted in figure 6(d) and figure 6(b) respectively. Evaluating the proposed method by comparing the other two protocols shows better efficiency for the proposed method due to the implementation of the machine learning strategy and its aim to reduce hop count; the proposed scheme outperforms 60% more than PRoPHET and about 47% more than PRoPHET+, as shown in figure 6(a). Similarly, due to its learning-based computation nature, the average latency performance of the proposed data transmission mechanism is acceptable.

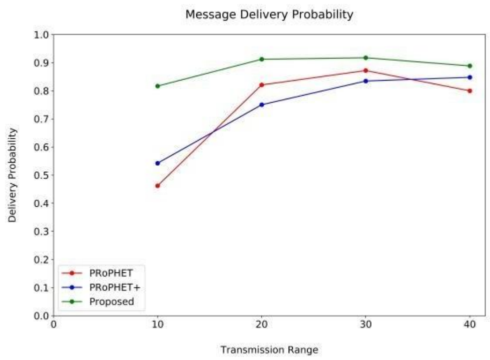

Figure 6(a): Transmission Range versus Delivery Ratio

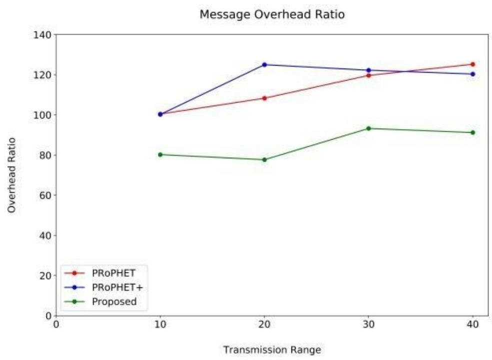

Figure 6(b): Transmission Range versus Overhead Ratio

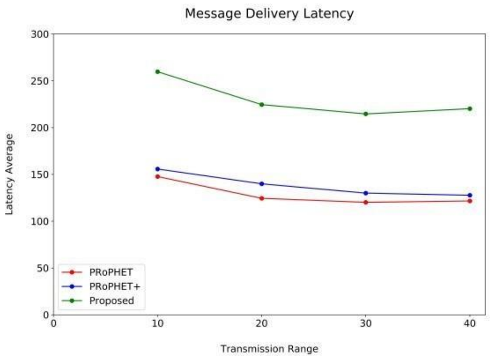

Figure 6(c): Transmission Range versus Latency Average

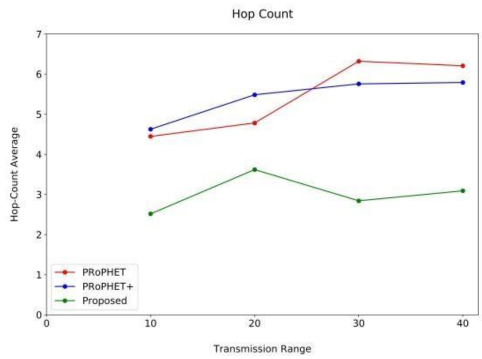

Figure 6(d): Transmission Range versus Hop Count Average

Evaluation with different Transmission Speed

Figure 7 shows evaluation of the proposed mechanism protocol by varying message transmission speed. When the transmission speed increases, it demands more data processing and computation speed. Due to intermittently connected networking resource limitations, the data handling speed requirement is impaired and eventually downgraded in the delivery probability, which is reflected in figure 7(d). Here at the beginning, while comparing with PRoPHET and PRoPHET+, it outperforms but for more computational demand in machine learning, it maintains a satisfactory level (delivery ratio > 80%). This downturn in delivery ratio also affects the overhead ratio, as shown in picture 7(b), still outperforming PRoPHET by 16% and PRoPHET+ by 15%. Increase in transmission speed also affects the hop count average metric but due to machine learning implementation, the increasing tendency of the slope is more controlled for the proposed protocol and outperforms the other two protocols, as presented in figure 7(a). Figure 7(c) shows that the proposed protocol holds a satisfactory latency average level regardless of its more constrained data routing strategy.

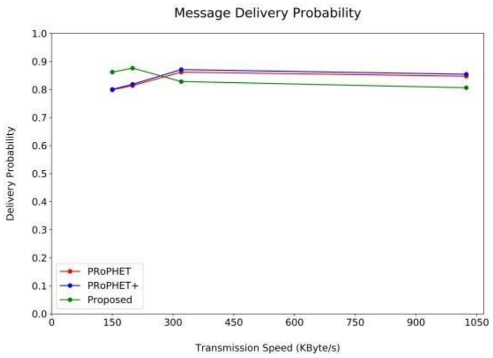

Figure 6(a): Transmission Speed versus Delivery Ratio

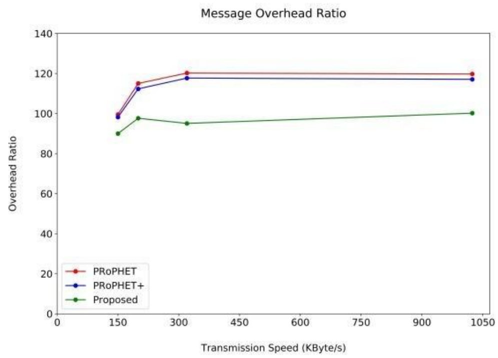

Figure 6(b): Transmission Speed versus Overhead Ratio

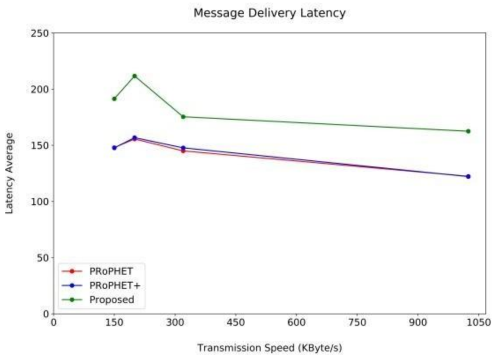

Figure 6(c): Transmission Speed versus Latency Average

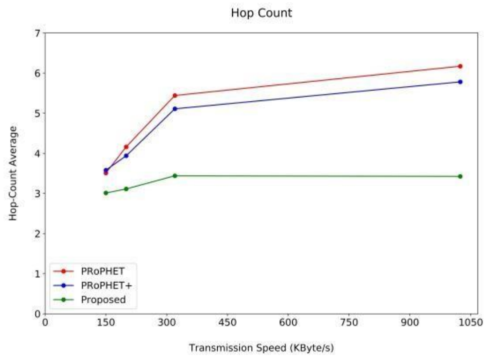

Figure 6(d): Transmission Speed versus Hop Count Average

5. CONCLUSION

To our knowledge, this is the first study of a machine learning application for predicting hop count. The data transmission method using the SVR predictor has demonstrated its success in efficient data forwarding with fruitful predictable hop count value for intermittently connected wireless sensor networks applied in the internet of healthcare-aware scenarios. After numerous experiments, it is evident that the performance of the data transmission mechanism with an SVR-based hop count value is significantly superior to that of PRoPHET and PRoPHET+, the other two baseline predictors in the same environment. Simulation results showed that the proposed protocol has a better hop count average, overhead ratio, and delivery probability, in most of the investigated parameter settings, despite a small average latency due to its machine learning computation-based hop count estimation strategy. The proposed protocol reduces the hop count average by approximately 50% compared to PRoPHET. It outperforms more than 40% of PRoPHET+, and the overhead ratio is reduced by approximately 20% than both the compared protocols by maintaining a delivery ratio > 80%. This evidences the applicability of SVR to determine hop count and thereby data forwarding mechanism using the same.

Future work should consider simulating the scheme in varying network environment parameters and other optimization methods to reflect the aspects of the intermittently connected networking environment as much as possible. Designing hybrid wireless data forwarding mechanism in the same environment should be evaluated in future works. The system also requires further improvement in several areas, like incorporating congestion-free data dissemination schemes, a load balancing approach, etc.

DECLARATION OF CONFLICTING INTEREST

The author(s) decleared no potential conflict of interest with respect to the research, authorship, and/or publication of this article.

ACKNOWLEDGEMENT

The first author would like to thank the Department of Science and Technology (DST), Government of India, for the INSPIRE Fellowship in support of this work.

REFERENCES

[1] Bujari, A.; Gaito, S.; Maggiorini, D.; Palazzi, C.E.; Quadri, C. Delay Tolerant Networking over the Metropolitan Public Transportation. Mob. Inf. Syst. 2016, 2016, 8434109.

[2] Han, SD, Chung, YW. An improved PRoPHET routing protocol in delay tolerant network. Sci World J 2015; 2015: 623090.

[3] Husni, E., Wibowo, A.: Delay tolerant network based e-mail system using trains. In: Proceedings of the Asian Internet Engineeering Conference (AINTEC 2012), pp. 17–22. November 2012

[4] D. D. Deb, S. Bose and S. Bandyopadhyay, "Coordinating disaster relief operations using smart phone/pda based peer-to-peer communication", International Journal of Wireless & Mobile Networks, vol. 4, no. 6, pp. 27, 2012.

[5] H. Wu, Y. Li and B. Cao, "Interplanetary communication technologies, architectures and applications," 2015 IEEE/CIC International Conference on Communications in China (ICCC), Shenzhen, 2015, pp. 1-6. doi: 10.1109/ICCChina.2015.7448645

[6] M. Malowidzki, P. Kaniewski, R. Matyszkiel and P. Berezinski, "Standard tactical services in a military disruption-tolerant network: Field tests", Military Communications Conference MILCOM 2017 - IEEE.

[7] A. Tovar, T. Friesen, K. Ferens and B. McLeod, "A dtn wireless sensor network for wildlife habitat monitoring", Electrical and Computer Engineering (CCECE) 2010 23rd Canadian Conference, pp. 1-5, 2010.

[8] M. Tasfe, B. Saha and A. Chakrabarty, "Social interest based pocket switched network communication", IEEE Region 10 Humanitarian Technology Conference, 2017.

[9] Gonçalves Filho, J., Patel, A., Batista, B. L. A., & Celestino, J. (2016). A systematic technical survey of DTN and VDTN routing protocols. Computer Standards & Interfaces, 48, 139–159.

[10] A. Vahdat, D. Becker, "Epidemic Routing for Partially-Connected Ad Hoc Networks", Duke Tech Report CS-2000-06, 2000.

[11] Spyropoulos, T., Psounis, K., and Raghavendra, C. S. Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks. In Proc. of the ACM SIGCOMM Workshop on Delay-Tolerant Networking (WDTN) (2005).

[12] A. Lindgren, A. Doria and O. Schelen, "Probabilistic routing in intermittently connected networks", SIGMOBILE Mobile Comput. Commun. Rev., vol. 7, no. 3, 2003.

[13] T. Huang, C. Lee, and L. Chen, "PRoPHET+: An Adaptive PRoPHET-Based Routing Protocol for Opportunistic Network," in 2010 24th IEEE International Conference on Advanced Information Networking and Applications, pp. 112-119, 2010.

[14] Maitreyi P, Rao MS (2017) "Design of Binary Spray and wait protocol for intermittently connected mobile networks." In 2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, pp. 1-3.

[15] Jingfeng Xue, Jiansheng Li, Yuanda Cao and Ji Fang, "Advanced PROPHET Routing in Delay Tolerant Network", 2009 International Conference on Communication Software and Networks IEEE, pp. 411-412.

[16] M. Sardari, F. Hendessi and F. Fekri, "Infocast: A new paradigm for collaborative content distribution from roadside units to vehicular networks", Proc. 6th Annu. IEEE Commun. Soc. Conf. SECON, pp. 1-9, 2009.

[17] T. Small and Z. J. Haas, "Resource and performance tradeoffs in delay-tolerant wireless networks", Proc. ACM SIGCOMM WDTN, pp. 260-267, 2005.

[18] X. Zhang, G. Neglia, J. Kurose and D. Towsley, "Performance modeling of epidemic routing", Comput. Netw., vol. 51, no. 10, pp. 2867-2891, Jul. 2007.

[19] X. Li, W. Shu, M. Li, H. Huang and M.-Y. Wu, "DTN routing in vehicular sensor networks", Proc. IEEE GLOBECOM, pp. 1-5.

[20] J. Burgess, B. Gallagher, D. Jensen and B. N. Levine, "MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks", Proc. 25th IEEE INFOCOM, vol. 6, pp. 1-11.

[21] P. Luo, H. Huang, W. Shu, M. Li and M.-Y. Wu, "Performance evaluation of vehicular DTN routing under realistic mobility models", Proc. IEEE WCNC, vol. 2, pp. 2206-2211.

[22] B. Balasubramanian, A. Levine and A. Venkataramani, "DTN routing as a resource allocation problem", Proc. Conf. Appl. Technol. Archit. Protocols Comput. Commun., pp. 373-384, 2007.

[23] F. C. Lee and C. K. Yeo, "Probabilistic Routing Based on History of Messages in Delay Tolerant Networks," 2011 IEEE Vehicular Technology Conference (VTC Fall), San Francisco, CA, 2011, pp. 1-6. doi: 10.1109/VETECF.2011.6093035

[24] S. K. Pandey and A. K. Singh, "Efficient prophet with buffer management for multicasting in DTN", Proc. IEEE Int. Conf. Inventive Res. Comput. Appl., pp. 1200-1205, 2018.

[25] V. N. Vapnik, The Nature of Statistical Learning Theory. New York: Springer, 2000.

[26] D. Bi, Y. F. Li, S. K. Tso and G. L. Wang, "Friction modeling and compensation for haptic display based on support vector machine," in IEEE Transactions on Industrial Electronics, vol. 51, no. 2, pp. 491-500, April 2004. doi: 10.1109/TIE.2004.825277

[27] C. Cortes and V. Vapnik, "Support-vector network", Machine Learning, vol. 20, pp. 273-297, 1995.

[28] C.-C. Chang and C.-J. Lin, LIBSVM: A Library for Support Vector Machines, 2001, [online] Available: .

[29] The Opportunistic Network Environment Simulator. Available online: https://akeranen.github.io/the-one/ (Accessed on 15 August, 2020)

[30] A. Keränen, J. Ott and T. Kärkkäinen, "The ONE simulator for DTN protocol evaluation", Proc. Int. Conf. Simultools Tech., pp. 55, 2009.

[31] F. Pedregosa et al., "Scikit-learn: Machine learning in Python", J. Mach. Learn. Res., vol. 12, pp. 2825-2830, Feb. 2011.

[32] S. Saha, A. Sheldekar, A. Mukherjee and S. Nandi, "Post disaster management using delay tolerant network" in Recent Trends in Wireless and Mobile Networks, Berlin, Germany:Springer, pp. 170-184, 2011.

[33] T. Spyropoulos, K. Psounis and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: The single-copy case", IEEE Trans. Netw., vol. 16, no. 1, pp. 63-76, Feb. 2008.

[34] Sagar Bose and Indrajit Banerjee "A Reliable Internet Of Things(IoT)Enabled Smart Healthcare Decision-Aid Method and System to Support Medical Management Services in Post Disaster Scenarios“ Indian Patent Filed, File No. 201831020949, 2018.

[35] A. Smola and B. Sch¨olkopf, “A tutorial on support vector regression,” Neuro COLT Technical Report TR-1998-030, Royal Holloway College, London, UK, 1998.