International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075, Volume-9 Issue-1S, November 2019

Machine Learning in Delay Tolerant Networks: Algorithms, Strategies, and Applications

R. Amirthavalli, S. Thanga Ramya

Abstract*: Delay Tolerant Networks (DTNs) has intermittent connectivity, nodes in the network experience a long delay in the delivery of packets, and the nodes are sparsely distributed. DTN is deployed in the applications where human intervention is least like underwater communication, interplanetary communication, disaster management, tracking wildlife, etc. Any changes in the environment affect the deployed sensor nodes, so it is required that the sensor nodes adapt to these environmental changes. Machine-Learning (ML) techniques can be deployed to overcome such difficulty. ML improves the network lifetime. ML in DTN facilitates routing by adapting to the network changes, mitigates congestion, reduces overhead. This paper provides a survey of ML techniques used in DTN. To the best of our knowledge, this work is the first of its kind to survey ML techniques in DTN.

Keywords*: Delay Tolerant Networks, Machine Learning, Routing, Supervised Learning.

I. INTRODUCTION

Delay Tolerant Networks (DTNs), as the name suggests, the sensor nodes in the network suffers long delay in the delivery of packets due to the intermittent connectivity. There is no end-to-end connectivity among the communicating nodes like WSNs due to environmental factors. So the sender stores the packet in the buffer till it finds the next intermediate node towards the destination. The intermediate node does the same until the packet is delivered to the destination. DTN was developed for interplanetary communication and later found its applications in mobile networks, underground networks, underwater acoustic networks, disaster management, wildlife tracking etc.,.

The routing in traditional networks, first establishes the routes among the communication parties, before the actual communication starts, which ensures reliable connections. But in the case of DTN, these networks lack end-to-end routes as nodes are sparsely distributed. So the traditional network protocols cannot be used in DTN. DTNs use Store-Carry-Forward approach [1] ie., the node stores and carries the packet until it is forwarded to the suitable next hop or destination.

ML is an application of Artificial Intelligence, which automates the system to self learn and update itself from past experiences. Nowadays, there is a huge surge in the number of applications that implement ML to automate, solve everyday activities. ML applications include image recognition, medical diagnosis, classification, prediction, etc.

The remaining paper is organized as follows: Section II provides information about Machine Learning Algorithms in Delay Tolerant Networks, Section III Challenges in DTN, Section IV draws conclusion on the description of various machine learning algorithms applied in DTN.

II. MACHINE LEARNING ALGORITHMS IN DELAY TOLERANT NETWORKS

Generally, machine learning is identified as an approach that helps in prediction, classification of data. Nevertheless, there are many research opportunities for machine learning techniques to be applied in DTN like routing, congestion control, to reduce overhead, etc.

Machine learning is broadly classified into supervised, unsupervised and reinforcement learning. In supervised learning, the system model is trained using the labeled dataset. Unsupervised learning builds the model without the need for the labeled training dataset. The similarity among the data in the dataset is considered to build the model. Reinforcement learning trains the agent to learn from the interaction with the environment.

A. Supervised Learning

This is a machine learning technique that trains the system model using a sample dataset to label test data. The input of the sample dataset is mapped to the corresponding labels, it belongs to. When this dataset is iterated in the system, the model learns to classify the test data to its corresponding labels.

K-Nearest Neighbor(k-NN): With the labeled training dataset, a sample data can be labeled to its near data sample. For continuous data, the Euclidean distance algorithm is used whereas, for discrete data Hamming distance is used to approximate to its near neighbor.

Based on the k-Nearest neighbor algorithm, [2] the routing decision is made with the following attributes are considered: the time index in the epoch, the source and the destination nodes, is the packet delivered or not?

Neural Networks: This supervised learning algorithm has many layers, with each layer having many nodes chained with the nodes in the previous and the next layers [3]. The first layer is the input layer whereas the last layer, the output layer. The intermediate layers are called hidden layers. The interconnections or chaining of the nodes represent the weight that impacts the nodes on the next layer. Each node in the hidden layer will have an activation function which dictates how active this node will be.

Revised Manuscript Received on November 22, 2019.
* Correspondence Author
Ms. R. Amirthavalli, CSE, Velammal Engineering College, Chennai, India. Email: amirthavalli@velammal.edu.in
Dr. S. Thanga Ramya, IT, RMD Engineering College, Chennai, India. Email: str.it@rmd.ac.in

In this paper [4], MLProph(Prophet routing algorithm based on Machine Learning) algorithm uses neural networks, which take 12 input parameters(like popularity parameter, buffer capacity, node energy, hop count, successful packet delivery rate, etc) and produces two outputs(successful delivery

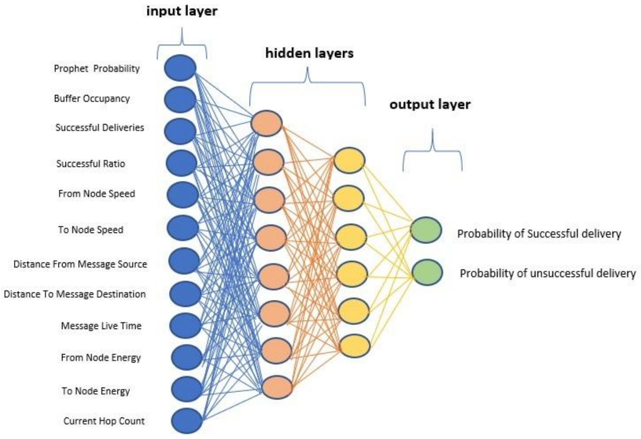

Fig. 1.A Sample Neural Network

probability and unsuccessful delivery probability) in the output layer. Fig.1. shows the sample Neural Network. The probability of successful delivery is based on the input values, given to the 12 input factors. In a neural network, the value computed at each node is a function of the linear combination of the nodes' values in the previous layers. Datasets are trained to get appropriate weights for a node in one layer to every other node in the next layer. The Backpropagation method is used to train datasets.

The next node and contact time is predicted, based on Artificial Neural Networks(ANN) [15], [16]. This neural network has 5 input parameters(current node id, contact begin time, contact end time, previous node id, previous contact begin time), one hidden layer and outputs, namely, Next contact node id, and next contact begin time. The Backpropagation method is used for training.

In paper [21], Trust Based Intelligent Routing(TBIR) deploys the ANN that calculates and learns the trust values of the nodes in the network which aids in making an intelligent routing decision. The trust function is based on the following three parameters, namely, (i) Time Difference between Recent Connection and Last Connection (ii) Frequency of Calls (iii) Total Duration.

In this paper [24], the liquid state machines(LSM), a type of reservoir computing, using the spiking neural networks(SNN), computes the optimal next hop while routing. The Leaky-Integrate-and-Fire (LIF) neuron model is deployed to determine the behavior of each neuron. Random one-fifth neurons are selected as inhibitory and others are excitatory. Input to this SNN is 10 neurons whereas there is a single output, that is the optimal next hop.

Decision Tree: Decision trees are a type of classification algorithm which predicts and labels when data is iterated through a decision tree [5]. While iterating the data through the decision tree, the factors or features of the dataset are compared in a decision unit, by considering the conditions required to branch to the subsequent child, which leads to a specific label or category for the data in question. Proper pruning of the decision tree is required so that the training data does not overfit affecting the classification. Fig. 2. shows the sample decision tree.

The MLProph, [4] when using a decision tree, uses 12 input attributes that classify the probability of successful delivery from unsuccessful deliveries. From the root node, each node in the decision tree either labels the output or leads to the next decision node based on the conditions. The decision tree is built recursively using the training data.

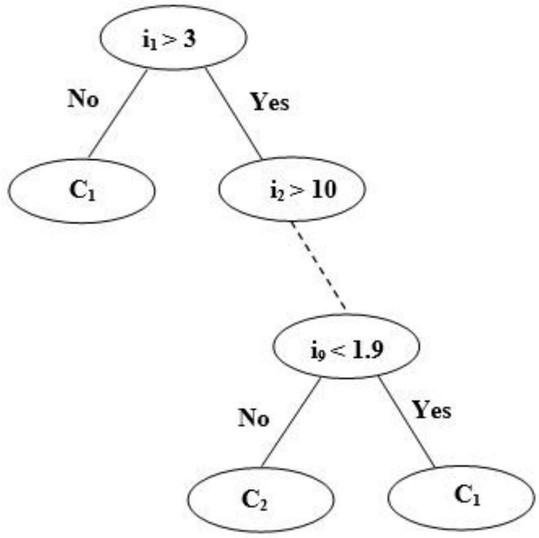

Fig. 2.A simple Decision Tree

In [6], the decision tree classification is applied while routing to decrease the overhead involved in DTN vehicular networks. The input attributes that build the decision tree are node ID, region code, lobby index, reception time, time passed and distance. Based on these values, the classifier decides whether to forward a packet to a neighbor or not.

In this paper [23], a new data dissemination algorithm, in delay tolerant networks, is proposed which considers the following factors, centrality, similarity, social strength, friendship, and trust for optimization to improve the forwarding probability. The decision tree provides the selection procedure to select the optimized values for the above factors that are adequate for target performance. It aids the selection of values for the parameters that satisfy the required target.

Bayesian Classifier: It is a simple, supervised learning algorithm that requires a small number of training samples to infer the classification. If A and B are two events, Bayes theorem states that the posterior probability of A, $P(A|B) \propto P(A)P(B|A)$, where $P(A)$ is the prior probability of A and $P(B|A)$ is the probability of observing A given B [7]. The only limitation of this classifier is that it requires prior distribution.

In [8] "FriendShip and Acquaintanceship Forwarding"(FSF) protocol, the social ties of the nodes in the network are classified into friends, acquaintances, and unknowns using the Naive Bayes Classifier. A database built from the MIT research experiment is used, which contains attributes like the number of meetings, meetings outside the University, In [9], a Bayesian classifier classifies the nodes as "delivered" and "non-delivered", for next hop towards the destination, by analyzing the input attributes, region code, and time slot, based on which the posterior probability will be calculated. Affiliation index is calculated, using Bayes Theorem, whose value decides the node in question will be chosen as the next hop.

In [10], the Naive-bayesian approach is used to predict the reliability of a neighboring node

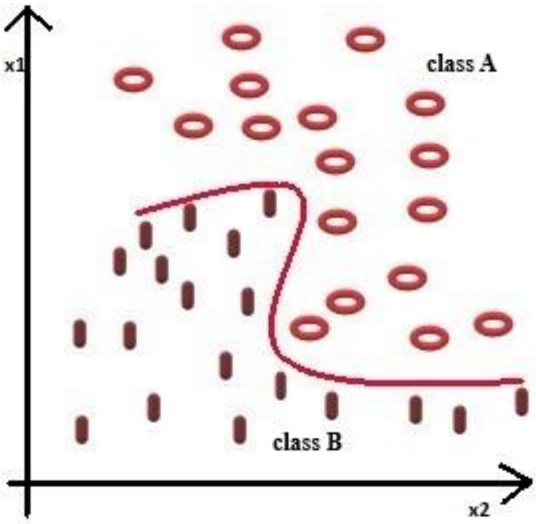

Fig. 3.Example Support Vector Machine

based on contact attributes like retransmission requests, average incoming data rate, outgoing data rate, distance, bit error rate, etc.

Support Vector Machine: This algorithm, a supervised training method, classifies data using a labeled training data set [19]. Fig. 3. shows an example SVM. SVM is trained to detect malware in delay tolerant networks. The Naive-Bayes trained sample data set is used to train the SVM [20].

B. Unsupervised Learning

It is a type of learning which classifies the data based on the similarity. This learning does not involve any training.

K-Means Clustering: The k-means algorithm clusters the data in the dataset based on the similarity [11]. The following are the steps in clustering: (i) Assume randomly k nodes to be the centroid of the clusters. (ii) for every node, using distance function, compute the nearest centroid. (iii) the new centroids are determined based on the current node's memberships (iv)halt the process if the convergence is reached or go to step (ii)

In [2], uses the k-means clustering algorithm to identify the regions which the nodes in the network visit often. Based on this clustering, the nodes are grouped for further classification.

Principal Component Analysis: It compresses multivariate data and reduces the dimensions by transforming correlated data into orthogonally, uncorrelated variables that are called principal components [17].

Post-disaster, principal components regression(PCR) forecasts the emergency resource needs in a shelter, based on the factors influencing the emergency [18]. Applying PCA, the influencing situational parameters(ISP) are selected and transformed into principal components(PC), such that the elements are uncorrelated.

C. Reinforcement Learning

This is a type of machine learning which steers the software agents to maneuver actions that maximize the reward in the given environment.

Q-Learning: [12] Q stands for "Quality", meaning a quality, optimal action-selection policy which dictates the action to be taken by the agent to maximize the rewards, for a given state. Markov Decision Process(MDP) defines Q-learning by specifying the State(S), Actions(A), state

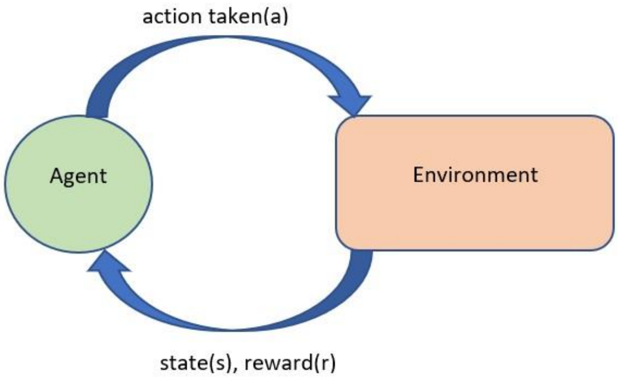

Fig. 4.Q-Learning Process

transition Probabilities(P) and Rewards(R) collected for a given state and taking an optimized action, generally referred as a tuple (S, A, P, R). In Q-learning, exploration and exploitation are the ways for the nodes to choose the action-selection strategy. Fig. 4. shows the Q-learning process. While exploring the action selected is different from the current best action taken for the same state. Whereas in exploitation, the action selected is the same as the current best action to maximize the rewards. During the initial training, action selection uses exploration whereas, at the end of the training, it uses exploitation.

Q-learning-based DTN routing protocol(QDTR) [13] translates the underwater routing problem into an MDP process to implement the Q-learning algorithm. According to this translation, the network is considered as a system, the state is linked with the node that carries a particular packet. When a packet reaches a sink, then the state, linked with this node is the terminal state, the highest of all states. The routing process identifies the required action to reach the terminal state, such that the rewards earned are maximized. The reward function relies on some factors and the weights associated with these factors. The factors are the relative distance to the sink, node density of an area, and the residual energy of an encountered node. By changing the weights of the factors, the algorithm achieves to earn maximum rewards.

Smart-DTN-Congestion Control(CC) [14] tries to mitigate congestion by applying the Q-learning technique. In this algorithm, every node in the network maintains the q-values in a q-table for all possible states and all possible actions(a) for that particular state(s), and Q(s,a) the action, taken for that state. The estimated q-values represent the efficacy of the nodes' actions to control congestion. Based on the previous observations and influenced by the environment, this learning algorithm strives to find a policy, which will maximize the reward(here to mitigate the congestion). The Boltzmann probability distribution is used in the exploration of action selection policy, whereas Win or Learn Fast(WoLF) method is used for exploitation.

In [10], the Q-learning approach is applied to remember the best choice of a reliable link to the neighbors. It stores the retransmission efforts of a particular contact and adds more weight to this contact if it is reliable, so that this contact can be used for future communication for the destinations in that direction.

Double Q-Learning Routing(DQLR) [22], selects the next hop in a distributed way. DQLR decouples the selection and evaluation with two value functions ie., the double Q-Learning functions. Every message in every node of the network is associated with these two functions. For the next hop, the value of one function depends on the value of another function. The messages in the node learn the action-value functions and based on the optimal future rewards the next hop is determined for the current node, to pass on the message.

Delay Tolerant Reinforcement-Based(DTRB) [25] algorithm uses Multi-agent reinforcement learning to learn the routes and copy the messages that generate rewards. The nodes in the network exchanging control messages which contain information about the rewards offered for a packet exchange and the distance-table. Every node maintains the reward table which consists of Q-values(estimate of future rewards when an agent takes an action in a particular state). Initially, these values are zero. While routing the sender sends control messages to the neighbor nodes with the reward attached. Based on this offer, the node which received this control message updates its estimate with the estimate attached to the control message.The rewards are inversely proportional to the distance between the communicating nodes.

III. CHALLENGES IN DTN

A. Routing

In [9], When a node encounters one or more neighbors, the node calculates the affiliation index for every neighboring node. The Bayesian classifier is used to calculate the affiliation index. The neighbor with high affiliation towards the destination is chosen to be the next hop.

FSF [8] applies the Naive Bayes classifier to discover the social ties of the nodes by classifying the interaction as friends, acquaintances, and unknown. A node in FSF will forward the packet to the node which can be its friend or acquaintance. If the encountered node is unknown, then the node does not forward it to an unknown node.

MLProph [4] applied both neural networks(MLProph$_{NN}$) and decision tree method(MLProph$_{DT}$) to calculate the MLProbabilty, which is used to predict whether the encountered node is capable of delivering the packet successfully. Simulation results show that MLProph$_{NN}$ outperformed MLProph$_{DT}$.

QDTR [13] algorithm is an adaptive and energy-efficient routing algorithm which adopted Q-learning. Adjusting the weights of the input parameters help realise the maximum reward like energy efficiency, minimised end-to-end delay, improve the delivery rate and reduce the storage overhead.

This paper [10] applies a hybrid machine learning approach(Bayesian classifier and Q-learning) to Contact Graph Routing to adapt to network changes. The Naive Bayesian classifier is applied for the reliability prediction of neighbor nodes. Q-learning approach helps to exploit the reliable neighbor nodes to be used for future communication.

In [2], the next hop node is chosen from a set of neighbors by applying the following Supervised learning methods: the Naive Bayes method, Decision Tree method, and K-Nearest Neighbor method. Out of these, Decision Tree has performed well compared to the other two. K-means clustering algorithm is applied to cluster the regions which nodes visit very often.

In TBIR [21], the routing decision is based on the calculated trust value. If one of the intermediate nodes have high trust value towards the destination, then that intermediate node is chosen for routing. If the intermediate nodes have the same trust value, then the routing decision is based on the latest connection time.

Table- I: Summary of Publications Resolving Various DTN Challenges by the Adoption of Machine Learning Techniques

	Routing	Reduce Routing Overhead	Congestion Control	Demand Forecasting
K-Nearest Neighbor	[2]			
Neural Networks	[4], [15], [16], [21], [24]			
Decision Trees	[4], [23]	[6]		
Bayesian Classifier	[8], [9], [10]			
K-Means Clustering	[2]			
Principal Componenet Analysis				[18]
Q-Learning	[13], [22], [25]		[14]	

B. Reduce Routing Overhead

In [6], the decision tree classifies the neighbor nodes to *m* classes. To illustrate the reduction of overhead, two routing algorithms are applied. They are Epidemic and SaW. To reduce the overhead in epidemic routing, retransmission probability vector V is set. V_i is the probability to forward the packet to a node of class C_i. So for a high class node low probability is set, and for a low class node, the packet is transmitted with high probability. In SaW, a maximum of L copies can be sprayed. To reduce the overhead, the copies spread should be less than L. Only high class nodes are chosen for next hop and low class nodes are not considered for next hop.

C. Congestion Control

Smart-DTN-CC[14] exploits the local information to predict the level of congestion in a node. A node is said to be in one of these states: Congested, NCongested means Non-congested, DCongested means Decrease-congested, PCongested means Prospective-congested. When a node from Congested or PCongested state changes to NCongested or DCongested state, the reward value is positive, whereas the state of the node changes from NCongested or DCongested state to Congested or PCongested state, the reward value is negative. Reward values range from 1 to -1. Due to this, the system can adjust to the changing environment. Table- I summarises the various DTN challenges by the adoption of machine learning techniques

IV. CONCLUSION

Delay tolerant networks are different from traditional networks. So, there is a need to address a variety of problems in DTN. Machine Learning approaches[26] can be applied to adapt to network changes, efficiently route the packets, reduce overhead, congestion control. Machine Learning approaches are accurate in prediction, converge very fast, learn from experience will help conserve energy and resources in a resource-limited DTN. Unlike Wireless Sensor Networks, not much research had been taken place in DTN using machine learning. So there is much scope for machine learning algorithms to be explored in DTN. The following are the future work that machine learning can be used in DTN: Functional challenges in DTN like data aggregation, event detection and query processing, localization, Nonfunctional challenges like security, intrusion detection system, quality of service, fault Detection, data integrity, etc.

REFERENCES

1. Bulut, Eyuphan, Zijian Wang, and Boleslaw Karol Szymanski., "Cost-effective multiperiod spraying for routing in delay-tolerant networks.", IEEE/ACM Transactions on Networking (TON) 18.5
2. Dudukovich, Rachel, and Christos Papachristou. "Delay Tolerant Network Routing as a Machine Learning Classification Problem." 2018 NASA/ESA Conference on Adaptive Hardware and Systems (AHS). IEEE, 2018.
3. Bengio, Yoshua. "Learning deep architectures for AI." Foundations and trends® in Machine Learning 2.1 (2009): 1-127.
4. Sharma, D. K., Dhurandher, S. K., Woungang, I., Srivastava, R. K., Mohananey, A., & Rodrigues, J. J. (2016). A machine learning-based protocol for efficient routing in opportunistic networks. IEEE Systems Journal, 12(3), 2207-2213.
5. Ayodele, Taiwo Oladipupo. "Types of machine learning algorithms." In New advances in machine learning. IntechOpen, 2010.
6. Portugal-Poma, Lourdes P., Cesar AC Marcondes, Hermes Senger, and Luciana Arantes. "Applying Machine Learning to Reduce Overhead in DTN Vehicular Networks." In 2014 Brazilian Symposium on Computer Networks and Distributed Systems, pp. 94-102. IEEE, 2014.
7. Box, George EP, and George C. Tiao. "Bayesian inference in statistical analysis". Vol. 40. John Wiley & Sons, 2011, pp 2-4.
8. Souza, Camilo, Edjair Mota, Diogo Soares, Pietro Manzoni, Juan-Carlos Cano, Carlos T. Calafate, and Enrique Hernández-Orallo. "FSF: Applying Machine Learning Techniques to Data Forwarding in Socially Selfish Opportunistic Networks." Sensors 19, no. 10 (2019): 2374.
9. Ahmed, Shabbir, and Salil S. Kanhere. "A bayesian routing framework for delay tolerant networks." In 2010 IEEE Wireless Communication and Networking Conference, pp. 1-6. IEEE, 2010.
10. Dudukovich, Rachel, Alan Hylton, and Christos Papachristou. "A machine learning concept for DTN routing." In 2017 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE), pp. 110-115. IEEE, 2017.
11. Kanungo, Tapas, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu. "An efficient k-means clustering algorithm: Analysis and implementation." IEEE Transactions on Pattern Analysis & Machine Intelligence 7 (2002): 881-892.
12. Watkins, Christopher JCH, and Peter Dayan. "Q-learning." Machine learning 8, no. 3-4 (1992): 279-292.
13. Hu, Tiansi, and Yunsi Fei. "An adaptive and energy-efficient routing protocol based on machine learning for underwater delay tolerant networks." In 2010 IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, pp. 381-384. IEEE, 2010.
14. Silva, Aloizio P., Katia Obraczka, Scott Burleigh, and Celso M. Hirata. "Smart congestion control for delay-and disruption tolerant networks." In 2016 13th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), pp. 1-9. IEEE, 2016.
15. Segundo, Fabio Rafael, Jean-Marie Farines, and Eraldo Silveira e Silva. "Quasi-opportunistic contact prediction in delay/disruption tolerant network." In Global Information Infrastructure Symposium-GIIS 2013, pp. 1-6. IEEE, 2013.
16. Segundo, Fabio Rafael, Eraldo Silveira e Silva, and Jean-Marie Farines. "A DTN routing strategy based on neural networks for urban bus transportation system." Journal of Network and Computer Applications 64 (2016): 216-228.
17. Jolliffe, Ian T., and Jorge Cadima. "Principal component analysis: a review and recent developments." Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 374, no. 2065 (2016): 20150202.
18. Basu, Souvik, Siuli Roy, and Sipra DasBit. "A Post-Disaster Demand Forecasting System Using Principal Component Regression Analysis and Case-Based Reasoning Over Smartphone-Based DTN." IEEE Transactions on Engineering Management 66, no. 2 (2018): 224-239.
19. Christmann, Andreas, and Ingo Steinwart. "Support vector machines." (2008).
20. Rajiha, K. M., DC Joy Winnie Wise, and S. N. Ananthi. "Malware Detection Based on Behavior in Delay Tolerant Network Using Support Vector Machine."
21. Singh, Ajay Vikram, Vandana Juyal, and Ravish Saggar. "Trust based intelligent routing algorithm for delay tolerant network using artificial neural network." Wireless Networks 23, no. 3 (2017): 693-702.
22. Yuan, Fan, Jaogao Wu, Hongyu Zhou, and Linfeng Liu. "A Double Q-Learning Routing in Delay Tolerant Networks." In ICC 2019-2019 IEEE International Conference on Communications (ICC), pp. 1-6. IEEE, 2019.
23. Reina, D. G., Radu-Ioan Ciobanu, S. L. Toral, and C. Dobre. "A multi-objective optimization of data dissemination in delay tolerant networks." Expert Systems with Applications 57 (2016): 178-191.
24. Lent, Ricardo. "Routing in a Delay Tolerant Network with Spiking Neurons." In ICC 2019-2019 IEEE International Conference on Communications (ICC), pp. 1-6. IEEE, 2019.
25. Rolla, Vitor G., and Marilia Curado. "A reinforcement learning-based routing for delay tolerant networks." Engineering Applications of Artificial Intelligence 26, no. 10 (2013): 2243-2250.
26. S.Thanga Ramya et al "Software Defined Networking: A Paradigm Shift in Networking for Future, Emerging Trends and Applications", International Journal of Applied Engineering Research, Volume 13, Number 18 (2018), pp.13475 – 13481.

AUTHORS PROFILE

Ms. R. Amirthavalli completed UG, B.E., (CSE) in Madras University, with First Class Degree in 2001. Secured First Class with Distinction in PG M.E., (CSE), Anna University, Coimbatore. She is an Assistant Professor in the department of CSE and has a teaching experience of 13 years. She is currently pursuing her Ph.D., (ICE) in Anna University, Chennai. Her Research interests include Wireless Sensor Networks, Delay Tolerant Networks, Network Security.

Dr. S. Thanga Ramya, B.E, M.S(by Res), Ph.D, is an Associate Professor in the Department of Information Technology, since June 2008. She obtained her B.E., (CSE) from Dr.Sivanthi Aditanar College of Engineering and M.S by Research (ICE) from Anna University, Chennai. She has obtained her Ph.D in Information and Communication Engineering from Anna University, Chennai, in 2017. She has been in the teaching profession for the past 18 years and has handled both UG and PG programs. Her areas of interest include programming languages, database management and data mining. She has published 9 papers in various International Journals and Conferences. She has attended many workshops & FDPs sponsored by AICTE related to her area of interest. She has also published 4 books. She is the life member of ISTE.