# Performance Analysis of Prophet Routing Protocol in Delay Tolerant Network by using Machine Learning Models

Bonu Satish Kumar[1], Sailaja Vishnubhatla[2], Chevuru Madhu Babu[3], Prof. S. Pallam Shetty[4]
Department of Computer Science & Systems Engineering, Andhra University, Visakhapatnam, India[1, 4]
Department of Computer Science, Government Degree College, Ravulapalem, India[2]
Department of Computer Science, SVSSC Government Degree College, Sullurupet, India[3]

***Abstract*—Delay-Tolerant Networking (DTN) or Disruptive-Tolerant Networking comes under the category of networks that works without infrastructure wireless networks. DTN is one type of computer network that provides solutions for several applications. Delay tolerant network communications are networks that are accomplished by storing packets briefly in intermediate nodes till a certain time an end-to-end route is been re-setup or regenerated. This leads to thought as Delay Tolerant Networks. The paper presents the developed models using Artificial Neural Networks (ANN) and Convolutional Neural Networks (CNN) for predicting the best alpha, beta, and gamma parameters of Probabilistic Routing Protocol for Intermittently Connected Networks (PROPHET) protocol for delay tolerant networks. The first data set is generated using ONE simulator, and the generated data is analyzed using python panda's module. From the above dataset, 80% was used for training and the remaining 20% each has been used for testing and validation. The models were developed and tested using the r2 score for both models to predict alpha, beta, and gamma parameters. Based on the predicted parameters extensive experiments were done and it was found that the ANN model is better than the CNN model. The ANN model can predict optimum alpha, beta, and gamma whereas CNN Model failed to produce accurate prediction.**

***Keywords—DTN; ONE; Prophet; CNN; ANN***

## I. Introduction

Wireless Networks are dynamic. Nowadays Wireless Networks have become a part of life to communicate with others. Delay Tolerant Network (DTN) is a type of wireless communication to communicate data. DTN is a wireless network designed to operate efficaciously in severe conditions and over very massive distances, including space area communications [1]. DTN networks are proven the most beneficial network in deep space communications. Space communications are lengthily distances of millions and hundreds of miles. So routing the data between massive distances leads to delays in transferring data, records losses, and errors. Present communication technology is not enough to handle such problems. That is wherein delay tolerant networks are introduced [2, 3].

To increase the quality of data-transferring services in large communications machine learning models are developed to predict the parameters for given inputs. The main reason for choosing DTNs is it decreases the delay and increases the throughput of the network [4]. As routing is a tedious task in communication hence DTN networks are chosen. Machine Learning (ML) belongs to a Sub-Category of Artificial Intelligence (AI) which doesn't need to program explicitly and it takes the input as the data samples and creates its insights from the data [5]. Here, the author created models of Deep Learning (DL) which take input as throughput and delay and output alpha, beta, and gama values without simulation. Deep Learning is a part of Machine Learning where Neural Networks (NN) are used to solve machine learning problems [6]. Neural Networks require high computational power to train the model.

Two neural network models Convolution Neural Networks (CNN) and Artificial Neural Networks (ANN) are defined for prediction. CNN is an architecture that is used for deep learning algorithms, specifically in the processing of pixel data in image processing concepts and pattern recognition in computer-related vision. The CNN also is the feed-forward network that is widely used for routing and multiple communication network tasks [7, 8]. An ANN is formed from a group of linked units or artificial neuron nodes, which are the simple model of the biological brain. ANN is the capability of paralleled processing, working with incomplete data, and memory distribution [9, 10].

In the present work, the author created machine learning models which can predict the required values just by inputting the data without simulation. The proposed paper is arranged as follows: The introduction part introduces the delay tolerant networks of the prophet routing protocol by using two machine learning models such as artificial neural networks and convolutional neural networks. The literature survey part presents the existing system. The next section methodology and simulation deals with the methods used to implement the performance of the prophet routing protocol in DTNs and shows the output using ONE simulator. The results and simulation part presents the prediction values that came from the CNN and ANN models. Comparing the above two models one is more accurate for all models and selected the best one is ANN. After doing extensive experiments, it was found that the ANN model is better than the CNN model with an accuracy of 99%. The final section deals with the conclusion of the present work and proposed a future scope of this study.

## II. Literature Review

Delivery ratio and packet drop influence quality of service in the delay tolerant networks are found and explained [11]. They proposed a solution to maintain the quality of service in DTN. They said that providing quality service in wireless networks is a challenging task. They investigated that no researcher now talked about fairness in the delay-tolerant network to maintain the Quality of service.

Research [12] worked in DTN routing as an ML classification problem. The authors discussed a machine learning-based style to directing the route for delay tolerant networks (DTNs). They explained various machine learning classification techniques to forecast a set of adjacent nodes which have the maximum possible to distribute communication to an anticipated location based on the message of the past delivery info. Their results showed that ML classification is a workable technique to expect network traffic, reducing overhead in epidemic-based routing approaches. Their model is not concentrated on new data arriving; this is the drawback of their work.

The thought and design of a machine learning-based routing for delay-tolerant planetary systems were discussed [13, 14]. They stated that the methods of Bayesian learning and reinforcement learning are used to enhance the routing decisions. Cognitive communications prototypes are studying popular ML algorithms which may be used further to enhance the functionality of existing routing algorithms. This approach proved the adaptive aids established in opportunistic transmitting strategies. In delay-tolerant network algorithms, strategies, and applications they incorporated Machine-Learning methods that can be fixed to overwhelm such struggles [15]. ML enhances the network lifespan in delay tolerant network. ML works on DTN routing by adjusting to the network changes diminishing congestion, and cutting overhead. The authors did not work to enhance the quality of service parameters like throughput and delay.

Study [16] used the Support Vector Machines (SVM) concept in machine learning used to solve non-linear classification and regression. The network traffic predictive method proposed here with some results, they prepared a dataset by using it for testing and training. They found that their approach to finding correctness is better than detecting the attacked traffic. The inclusion of neural networks of machine learning in wireless sensor networks is a useful tool to increase network performances was suggested in [17]. Work was done on ML techniques for the prediction of routing parameters and protocol to obtain optimal performance [18]. Their results show that improvement in the quality of service has been achieved.

Extensible Provisioning Protocol (EPP) is proposed to consider specific physical attributes of collaborating mobile nodes, at the side of their positional attributes with a current message time-to-live value, and determine routing hops [19]. Their primary aim is to increase message delivery rate while keeping an optimal balance between hop count, routing overhead, and cooperation among nodes. Extensive provisioning protocol works on delivery rate, this predictability value is manipulated using a weighted function of parameters like bandwidth, power, nodes buffer, deliverable probability, and popularity. Extensive provisioning protocol maintains high probability delivery while structured overhead ratio average hop count and balancing average latency.

Cognitive Radio Mobile Ad-hoc Networks (CR-MANETs) are proposed as an efficient quality of service protocol, where a quality of service route is formed by exploiting Deep Reinforcement Learning (DRL) [20]. The higher Packet Delivery Ratio (PDR) and lower energy consumption provides by the proposed quality of service routing protocol than the Cognitive Radio Ad-hoc On-demand Distance Vector (CR-AODV). Moreover, the cognitive radio Ad-hoc on-demand distance vector protocol spends too much energy for the Route Request Packet (RREQ) flooding, while the quality of service routing protocol just uncast an RREQ packet to the best neighbor relying on the Deep Reinforcement Learning (DRL) model, thus consuming the small energy.

A systematic and comprehensive study of investigation on various modern approaches for intensifying security in Mobile Ad-hoc Networks (MANETs) is represented in [21]. A review on routing protocol and security attacks in mobile ad-hoc networks studied different security attacks in MANET. The author also examined how different layers under the protocol stack become vulnerable to different types of attacks [22]. A different routing protocol was designed for infrastructure-less networks and described the operation of each protocol and then compared their characteristics. They classified the routing method based on the operation [23]. An Ad hoc routing protocol of table-driven and on-demand category are selected for their study [24]. This paper studied table-driven routing protocols and listed their pros and cons in various situations. Results getting from the Taguchi approach, it is conferred that one ideal setting for one response metric is not similar to another metric. Hence multiple-response metric study may yield better results as it gives one optimum setting to enhance all metrics [25]. The data transfer between air-linked devices to the ground station using Micro Aerial Vehicle (MAVlink) protocol is explained [26]. The work describes the transfer between micro aerial vehicles or unmanned aerial vehicles to ground control stations in different circumstances.

They categorized various machine learning algorithms based on security approaches in mobile ad-hoc networks. The security approaches are divided into three dimensions (3D) ML Based Intrusion Detection System (IDS), trust-based models, and attack detection models. In the existing system people who worked on DTNs simulated the model to improve the network throughput and to decrease the delay. The above models are worked to increase the throughput of the delay tolerant networks using the simulation.

## III. Methodology and Simulation

### A. Artificial Neural Network

Artificial Neural Networks (ANNs) are simply called Neural Networks (NN), these are the computing systems that are inspired by human and animal brain process information [27]. ANNs accumulate their expertise utilizing detecting the patterns and trained similarities within the data through experience, not from programming. The architecture of ANN is shown in Fig. 1. An ANN is formed from a group of linked units or artificial neuron nodes, which are the simple model of the biological brain.

In the ANN, each neuron is connected with weights (Coefficients) to each one to transmit signals from each other and organized in layers [28]. The coefficients decrease and increase with the strength of the signal in some thresholds. Based on the threshold the signal passes through the neurons within the unit. The output of the signal (neurons) is calculated by the non-linear equation of the sum of its input.

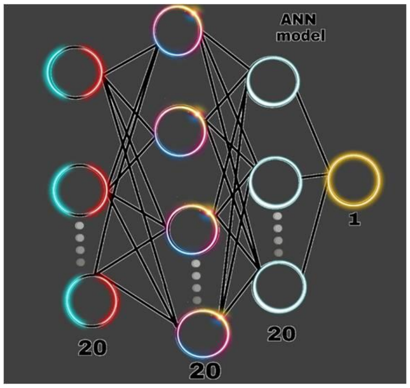


Fig. 1. Architecture of ANN.

## B. Convolutional Neural Networks

The problem with regular NNs is the lack of adaptability. At the same time complexity and size of the data to be increased, and the calculation power of the model should be increased this leads to more expensive neural networks. The parameter sharing is well known that the associated weight (Coefficient) in the CNN layers is remaining fixed. The parameter sharing in the CNN layer system will be less computationally intensive than the artificial neural networks. The architecture of ANN is shown in Fig. 2.

CNN is an architecture that is used for deep learning algorithms, specifically in the processing of pixel data in image processing concepts and pattern recognition in computer-related vision [29]. CNN also shows connectivity pattern similar to a human brain just like ANN and it consists of a group of linked neurons to connect. The CNN delivers better performance with image inputs, and also with speech or audio signal inputs.

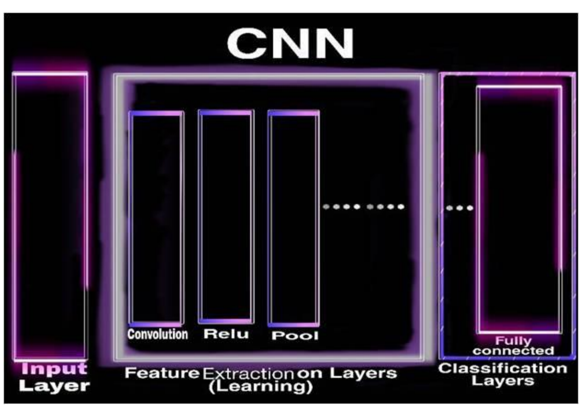


Fig. 2. Architecture of CNN.

## C. Simulation

A simulation is a duplicate model that mimics the operation of a current or proposed work, imparting proof for decision–making by having the ability to test distinct situations or different scenarios or method modifications, or process changes. A network system is a set of network elements, including switches, routers users, links, and packages operating collectively to acquire some assignments. The scope of a simulation examination, can also most effectively be a system inside another system as in the case of sub-networks.

The state of a network system is the set of suitable parameters and variables that narrate the system at a precised time that incorporates the scope of the study. As an example, if the interest is in the usage of a link, then recognize the quantity of bits transmitted through the link in a second and the full capacity of the link, instead of the variety of buffers to be available for the ports in the switches connected via link.

*1) Types of simulators:* Different network simulators offer different features.

- Network Simulator2.
- Network Simulator3.
- OPNET.
- OMNeT++.
- NetSim.
- ONE.
- REAL.
- QualNet.
- J-Sim.

*2) ONE Simulator:* DTN is a communication networking pattern that enables communication in environments wherein there may be no end-to-end paths, communication opportunities come and cross and their programming languages can be very lengthy and now not even identified beforehand. Routing messages in this type of surroundings may be unique and different from traditional standard networks. This gap has created a need to find different types of new routing protocols that take efficaciously into consideration the distinct nature of these networks.

Distinct procedures may be tested and evaluated by way of simulation. So in this work, the author considers and thinks about Opportunistic Network Environment (ONE) Simulator shown in Fig. 3. Not like other delay tolerant networks simulators, which typically pay attention to routing simulation, the ONE combines delay tolerant network routing, mobility modeling, and visualization in one package deal that is effortlessly extensible and offers a rich set of reporting and studying modules. The ONE simulator is a simulation tool that is used widely by researchers operating on studies related to opportunistic and DTN networks. The default_settings.txt notepad file is essential for running or compiling the stipulated simulation.

Standardize the simulation's configuration parameters along with alpha, beta, and gamma parameters from the Internet Engineering Task Force (IETF) draft of the Prophet routing protocol wherein static values are not suitable for delay tolerant networks; that are dynamic values shown in Table I. The working environment and parameters for the simulation setup are shown in Table II.

TABLE I. IEFT DRAFT PARAMETERS FOR PROPHET ROUTING PROTOCOLS

| Parameter | Default Value | Description |
|---|---|---|
| Alpha | 0.75 | Predictability initialization constant |
| Beta | 0.25 | Delivery predictability transitivity scaling constant |
| Gamma | 0.98 | Predictability aging constant |

TABLE II. VARIABLES FOR SIMULATION SETUP

| Name of Parameter | Value |
|---|---|
| Time of Simulation | 0.5hr-12hr |
| Node Density | 40,80,120 |
| Update interval | 0.1 |
| Interface Type | Simple Broadcast |
| Interface | Bluetooth |
| Interface Transmit Speed | 2 Mbps |
| Transmit Range | 100m |
| Routing Protocols | PRoPHET |
| Message TTL | 300 minutes(5hours) |
| Message Generation Rate | 25sec-35sec |
| Buffer Size | 2 MB |
| Message Size | 500 KB-1 MB |
| Movement Model | Shortest Path Map Based |
| Simulation Area Size | 4500m×3400m |
| Load time | 1Mb,2Mb,3Mb,4Mb,5Mb |

## D. Graphical User Interface Mode

The Graphical User Interface (GUI) additionally maintains track of noted events or activities and disposes them inside the panel of the event log. By clicking a message or a node name within the panel of the event log, more statistics may be proven approximately that message or node. The panel of event log dominance panel may be used to alter which events or activities are shown within the event log and the simulation also can be made mechanically paused in the case of some sort of activities or events.

## E. Methodology

The overall algorithm for this work is as follows and the flow chart of the algorithm is shown in Fig. 4.

Step1. Read the data from the data set.

Step2. Processing the data.

Step3. Apply artificial neural network.

Step4. Apply convolution neural network.

Step5. Evaluate and Compare the results.

A Java simulator is used to create the data, and Anaconda software is used to perform machine learning algorithms. Many people used Fuzzy Models to enhance the performance of DTN but here we used Machine Learning Models.

The following steps are discriminating the proposed methodology:

1) *Reading the data:*

*a)* By using the Pandas module imported the training Data set and testing data which was stored in Comma Separated Values (CSV) format.

*b)* Here, two lists are created that contain labels and targets.

2) *Modifying data:*

*a)* Using the time step of 30 apply time-series.

*b)* Here 30 denotes the input size to the model which is used to predict the targets.

*c)* Perform the Min Max Operation on the data.

*d)* Split the data for training and testing purposes with an 80:20 ratios.

3) *Modeling:* Modeling presents the detailed procedure for this study using two models such as convolutional neural networks as model-1 and artificial neural networks as model-2. The procedure of the above two models is discussed in the results and discussions.

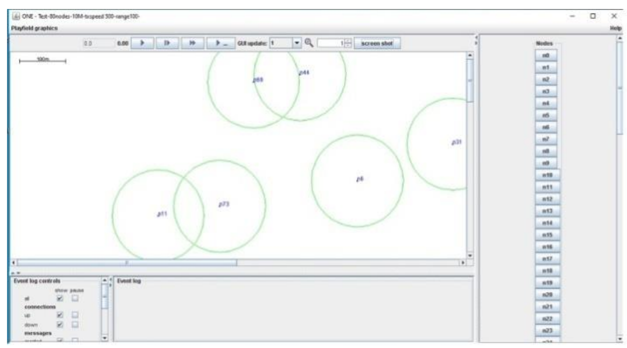

Fig. 3. ONE simulator GUI.

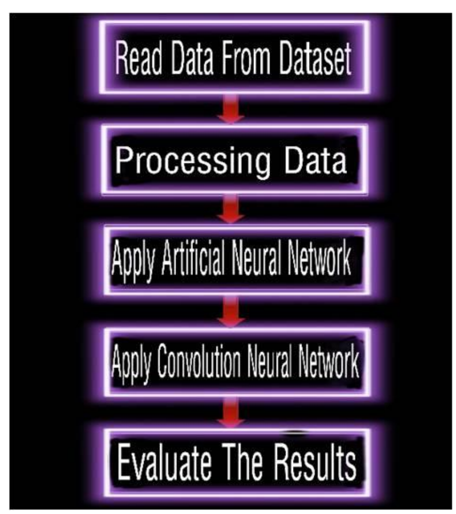


Fig. 4. Flow chart for the proposed system.

# IV. Results and Discussions

## A. Convolutional Neural Networks(Model-1)

### 1) Model-1 (CNN):

*a)* Import Sequential and Dense from Keras. models and keras.layers respectively.

*b)* Initialize the Sequential Model.

*c)* Add 4 dense layers with normal kernel initializer and Rectified Linear Unit (ReLU) activation function with 20, 25, 10, and 1 neurons in each respectively.

*d)* Compile the model with Adaptive Moment Estimation (ADAM) optimizer and Mean_Squared_Error Loss.

*e)* Fit the model with a validation split of 20% and run it for 70 epochs.

*f)* Make the Predictions and compare them with actual values shown in Table III.

*g)* Compute the r2 Score.

*h)* Finally, this model too failed to predict the output.

*i)* Now retrain the Conv1D Model with modified data for 100 epochs and a 20% validation split.

*j)* Compare the predicted values with the actual values and compute the r2_score.

*k)* The model worked successfully with 98% accuracy.

For this experiment, a total of five samples were used, and the samples are numbered 0, 1, 2, 3, and 4. Where the actual and predicted values obtained from the above CNN model are more similar to each other and their accuracy is approximately 98%.

TABLE III. PREDICTED AND ACTUAL VALUES OF THE CNN MODEL

| Value | Actual | Predicted |
|---|---|---|
| 0 | 0.835 | 0.820380 |
| 1 | 0.840 | 0.826132 |
| 2 | 0.850 | 0.832045 |
| 3 | 0.855 | 0.837852 |
| 4 | 0.860 | 0.843962 |

Added four dense layers with the return sequence of each one with 70 neurons and Epoch of 1/100, 2/100, and 3/100 for each 1 neuron add the dense layer to get the predicted output shown in Table IV. The training dataset of the CNN model is shown in Fig. 5.

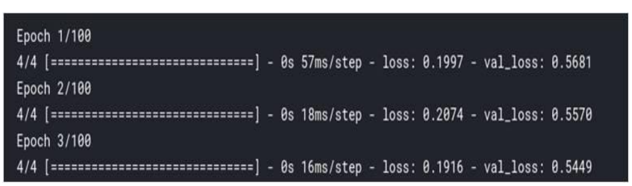


Fig. 5. The training data set of the CNN Model.

TABLE IV. DENSE LAYERS STEP EPOCH OF CNN MODEL

| Epoch | Step | Loss | Val_Loss |
|---|---|---|---|
| 1/100 (4/4) | 0s 57ms | 0.1997 | 0.5681 |
| 2/100 (4/4) | 0s 18ms | 0.2074 | 0.5570 |
| 3/100 (4/4) | 0s 16ms | 0.1916 | 0.5449 |


Fig. 6. Accuracy output of the CNN model.

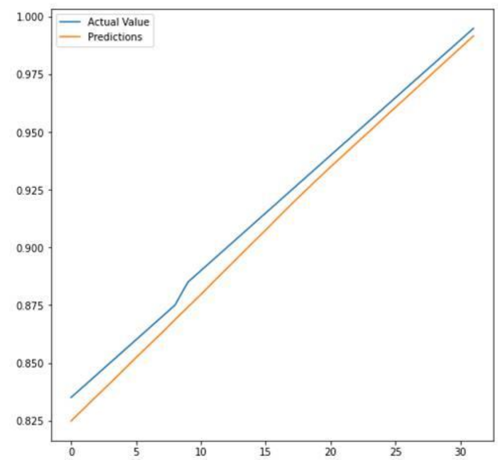


Fig. 7. Actual and predicted values of the CNN model.

Then retrain the Conv1D model with modified data for 100 epochs (1/100, 2/100, 3/100) and a 20% validation split. Compare the précised values with the actual values and compute the r2_score. The CNN model worked successfully with an accuracy of 98% shown in Fig. 6. The graph shown in Fig. 7 represents the accuracy CNN model with actual and predicted values. The yellow line shows the predictions and the blue one shows those actual values.

## B. Artificial Neural Networks(Model-2)

### l) Model-2 (ANN):

*a)* Import sequential and utils from TensorFlow.keras and import Flatten, Dense, Conv1D, MaxPool1D, and Dropout from TensorFlow. Keras. layers.

*b)* Initialize the Sequential Model.

*c)* Add 3 Conv1D Layers with kernel size (3) and ReLU activation function with 32, 64, and 128 neurons each.

*d)* Add a Flattening Layer.

*e)* Add three Dense Layers with ReLU activation function with 50, 20, and 1 neurons.

*f)* Compile the model with mean_squarred_error and Adam optimizer.

*g)* Fit the model with a 20% validation split for 100 epochs.

*h)* Predict the values using the model and compare with actual values and compute the r2_score shown in Table V.

*i)* This model too failed to predict the outputs and gave less accuracy.

*j)* Now retrain the Conv1D Model with modified data for 100 epochs and a 20% validation split.

*k)* Compare the predicted values with the actual values and compute the r2_score.

*l)* The model worked successfully with 99% accuracy.

TABLE V. PREDICTED AND ACTUAL VALUES OF THE ANN MODEL

| Value | Actual | Predicted |
|---|---|---|
| 0 | 0.835 | 0.830679 |
| 1 | 0.840 | 0.836426 |
| 2 | 0.845 | 0.842122 |
| 3 | 0.850 | 0.847810 |
| 4 | 0.855 | 0.853515 |

For this experiment, a total of five samples were used, and the samples are numbered 0, 1, 2, 3, and 4. Where the actual and predicted values obtained from the above ANN model are more similar to each other and their accuracy is approximately 99%.

The test data has been sent to the model to predict the output and the predictions have been stored in a variable as a data frame. The actual value and the predicted value have been compared to evaluate the model.

Add 3 Conv1D Layers with kernel size (3) and ReLU activation function with 32, 64, and 128 neurons each and Epoch of 1/100, 2/100, and 3/100 for each increasing neuron add the dense layer to get the predicated output shown in Table VI. The training dataset of the ANN models is shown in Fig. 8.

```
Epoch 1/100
4/4 [==============================] - 1s 72ms/step - loss: 0.2069 - val_loss: 0.2005
Epoch 2/100
4/4 [==============================] - 0s 20ms/step - loss: 0.0488 - val_loss: 0.1579
Epoch 3/100
4/4 [==============================] - 0s 20ms/step - loss: 0.0362 - val_loss: 0.0081
```

Fig. 8. Training data set in ANN Model.

TABLE VI. DENSE LAYERS STEP EPOCH OF ANN MODEL

| Epoch | Step | Loss | Val_Loss |
|---|---|---|---|
| 1/100 (4/4) | 1s 72ms | 0.2069 | 0.2005 |
| 2/100 (4/4) | 0s 20ms | 0.0488 | 0.1579 |
| 3/100 (4/4) | 0s 20ms | 0.0362 | 0.0001 |

Now retrain the Conv1D Model with modified data for 100 epochs and a 20% validation split. Compare the predicted values with the actual values and compute the r2_score. The model worked successfully with 99% accuracy shown in Fig. 9.

```
Accuracy score of the predictions: 0.9897303879778176
```

Fig. 9. Accuracy output of the ANN model.

The actual values of the data and the predicted values of the data have been plotted into a graph to check the variance between values shown in Fig. 10.

After the predicate of the actual and predicted values using ANN and CNN models the final output should be shown in Table VII. The actual values of the data and the predicted values of the data have been differentiated with equal intervals of 0.10, 0.01, 0.01, 0.009, and 0.009.

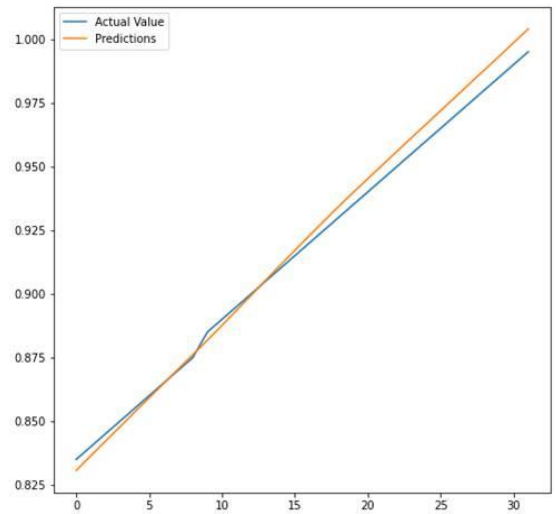


Fig. 10. Actual and predicted values of the ANN model.

TABLE VII. PREDICTED VALUES OF THE CNN AND ANN MODELS

| Value | CNN Predicted | ANN Predicted |
|---|---|---|
| 0 | 0.820380 | 0.830679 |
| 1 | 0.826132 | 0.836426 |
| 2 | 0.832045 | 0.842122 |
| 3 | 0.837852 | 0.847810 |
| 4 | 0.843962 | 0.853515 |

## V. CONCLUSION AND FUTURE SCOPE

Prophet routing protocol is one of the extensively studied delay-tolerant network protocols. In probabilistic routing protocol for intermittently linked networks, a message is forwarded to a touch node if the touch node has a better delivery quality of being expected to the destination of the message. A delay tolerant network is a Prophet routing protocol that gives confident delivery of facts using automatic store and forward mechanisms. Delay tolerant network comes under the category of networks that works without infrastructure wireless networks. Each data packet of transmission is received and immediately forwarded if possible, but these are stored for further transmission if forwarding is not possible currently however is predicted to be possible in future transmission.

In this research work, ANN and CNN models are used to predicate the best alpha, beta, and gamma parameters of prophet routing protocol for delay tolerant networks using a dataset simulated by ONE simulator. While considering models like CNN, ANN performed well on the test dataset. Of the dataset, 80% was used for training and the remaining 20% each has been used for testing and validation. While comparing the two models (CNN and ANN) the impact of CNN is mainly on the dynamic values of all datasets with an accuracy of 98% and ANN is mainly on the dynamic values of all datasets with an accuracy of 99%. Comparing the above two models one is more accurate for all models and selected the best one is ANN with 99%. After doing extensive experiments, it was found that the ANN model is better than the CNN model with an accuracy of 99%. At the same time, the ANN model can predict optimum alpha, beta, and gamma whereas the CNN model failed to produce accurate prediction.

The future scope should be trying to simulate a large number of datasets and trying to use predefined models like VGG-16, ResNet, and Efficient Net. Also considering the other parameters of DTN Routing Protocols to predicate the accurate results.

## REFERENCES


[1] Cerf, V., Burleigh, S., Hooke, A., Torgerson, L., Durst, R., Scott, K., ... & Weiss, H. (2007). Delay-tolerant networking architecture (No. rfc4838).

[2] Caini, C. (2021). Delay-tolerant networks (DTNs) for satellite communications. In Advances in Delay-Tolerant Networks (DTNs) (pp. 23-46). Woodhead Publishing.

[3] Mallorquí, A., Zaballos, A., & Serra, D. (2022). A Delay-Tolerant Network for Antarctica. IEEE Communications Magazine, 60(12), 56-62.

[4] Verma, A., & Kumar, S. (2021). Routing protocols in delay tolerant networks: Comparative and empirical analysis. Wireless Personal Communications, 118(1), 551-574.

[5] Zhou, Z. H. (2021). Machine learning. Springer Nature.

[6] Janiesch, C., Zschech, P., & Heinrich, K. (2021). Machine learning and deep learning. Electronic Markets, 31(3), 685-695.

[7] Park, D. C., & Choi, S. E. (1998, May). A neural network-based multi-destination routing algorithm for communication networks. In 1998 IEEE International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98CH36227) (Vol. 2, pp. 1673-1678). IEEE.

[8] Steur, N. A., & Schwenker, F. (2021). Next-generation neural networks: Capsule networks with routing-by-agreement for text classification. IEEE Access, 9, 125269-125299.

[9] Wang, C. J., & Weissler, P. N. (1995). The use of artificial neural networks for optimal message routing. IEEE Network, 9(2), 16-24.

[10] Lu, J., Li, D., Wang, P., Zheng, F., & Wang, M. (2022). Security-aware routing protocol based on artificial neural network algorithm and 6LoWPAN in the internet of things. Wireless Communications and Mobile Computing, 2022, 1-8.

[11] Roy, A., Acharya, T., & DasBit, S. (2018). Quality of service in delay tolerant networks: A survey. Computer Networks, 130, 121-133.

[12] Dudukovich, R., & Papachristou, C. (2018, August). Delay tolerant network routing as a machine learning classification problem. In 2018 NASA/ESA Conference on Adaptive Hardware and Systems (AHS) (pp. 96-103). IEEE.

[13] Dudukovich, R., Hylton, A., & Papachristou, C. (2017, October). A machine learning concept for DTN routing. In 2017 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE) (pp. 110-115). IEEE.

[14] Bajpai, S., & Chauhan, A. (2022, December). Evolution of Machine Learning Techniques for Optimizing Delay Tolerant Routing. In 2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N) (pp. 294-299). IEEE.

[15] Singh, A. K., & Pamula, R. (2021). Vehicular delay-tolerant network-based communication using machine learning classifiers. In Architectural Wireless Networks Solutions and Security Issues (pp. 195-208). Springer, Singapore.

[16] Sharma, A. K., & Parihar, P. S. (2013). An effective dose prevention system to analyze and prediction of network traffic using support vector machine learning. International Journal of Application or Innovation in Engineering & Management, 2(7), 249-256.

[17] Barbancho Concejero, J., León de Mora, C., Molina Cantero, F. J., & Barbancho Concejero, A. (2006). Giving neurons to sensors. QoS management in wireless sensors networks. In ETFA 2006: IEEE Conference on Emerging Technologies and Factory Automation (2006), p 594-597. IEEE Computer Society.

[18] Zafar, M. H., & Altalbe, A. (2021). Prediction of Scenarios for Routing in MANETs Based on Expanding Ring Search and Random Early Detection Parameters Using Machine Learning Techniques. IEEE Access, 9, 47033-47047.

[19] Samanta, R. EPP: Enhanced PRoPHET+ an efficient, cooperative, and network attributes based dtn routing protocol.

[20] Nguyen, T. V., Tran, T. N., Huynh-The, T., & An, B. (2021). An efficient QoS routing protocol in cognitive radio MANETs: Cross-layer design meets deep reinforcement learning. In 2021 International Conference on Electronics, Information, and Communication (ICEIC) (pp. 1-4). IEEE.

[21] Popli, R., Sethi, M., Kansal, I., Garg, A., & Goyal, N. (2021, August). Machine Learning Based Security Solutions in MANETs: State of the art approaches. In Journal of Physics: Conference Series (Vol. 1950, No. 1, p. 012070). IOP Publishing.

[22] Poonam Sihag, Saurabh Charaya, et al., in their paper entitled, “Routing Protocol & Security Attack in MANET: A Review”, 2018 IJCRT, Volume6, Issue 2 April 2018, ISSN: 2320-2882.

[23] E.M Royer, R.Noyer and C.K.Toh, “A Review of Current Routing Protocols for Ad hoc mobile wireless Networks”, Personal Communications, IEEE, Volume: 6, Issue: 2, pages 261-271, 1999.

[24] Hemagowri J., Baranikumari C., Brindha B., “A Study on Proactive Routing Protocol in Ad-hoc network”, International Journal of Modern Engineering Research, National Conference on Architecture, Software Systems and Green Computing (NCASG), 2019, pg.01-04.

[25] Swati Saxena, Madhavi Sinha,” Single-Response metric analysis of Adaptive Fault Tolerant Replication Routing Protocol for MANETs using Taguchi Approach”, International Journal of Computer Networking, Wireless and Mobile Communications (IJCNWMC), Volume. 3, Issue 3, Aug 2013, pages. 105-116.

[26] Mogili, U. R., & Deepak, B. B. V. L. (2020). An intelligent drone for agriculture applications with the aid of the MAVlink protocol. In Innovative Product Design and Intelligent Manufacturing Systems: Select Proceedings of ICIPDIMS 2019 (pp. 195-205). Springer Singapore.

[27] Hardesty, Larry (14 April 2017). "Explained: Neural networks". MIT News Office. Retrieved 2 June 2022.

[28] Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2021). A survey of convolutional neural networks: analysis, applications, and prospects. IEEE Transactions on neural networks and learning systems.

[29] Yegnanarayana, B. (2009). Artificial neural networks. PHI Learning Pvt. Ltd.