

Deep Learning for Wireless Network Handoff Management

Posted on March 13, 2025

Abstract

Handoff events, in which mobile nodes switch between various network access points, frequently result in performance degradation in wireless networks. TCP-PRN (Path Recovery Notification), a method designed to lessen the negative effects of temporary link disconnections, was developed in response to this challenge. In spite of its potential, precise handoff prediction and managing link disconnections continue to be major obstacles in actual deployment. This study introduces an improved TCP-PRN mechanism that combines ensembled deep learning methods with hybrid optimization techniques. These methods include Convolutional Neural Networks (CNNs), Stacked Autoencoders (SAEs), and reinforcement learning via Deep Networks (DQN). The fundamental changes to sender and receiver operations required by our improved TCP-PRN's adaptive congestion control and deep learning-based handoff prediction are the main parts of the system. A CNN-based approach aids in accurate handoff event prediction, facilitating proactive packet recovery strategies. Meanwhile, SAEs help in feature extraction from network data, improving prediction precision. The implementation of DQN makes it possible for senders to behave in an adaptive manner that is influenced by expected handoff events and current network congestion conditions. Additionally, we use hybrid optimization methods to improve TCP-PRN congestion control decisions by combining the Tuna Swarm with the Whale Optimization (Tuna customized whale optimization) algorithm. The effectiveness of our improved TCP-PRN mechanism in realistic environments is conclusively shown by simulation results, particularly in circumstances where handoff prediction presents a significant challenge. In addition to improving network reliability, the combination of deep learning and hybrid optimization ensures flexibility and effectiveness during handoff events. This method will ultimately improve the robustness and adaptability of wireless network protocols.

Introduction

Mobile wireless technology has experienced incredible growth due to its capacity to give users on the go constant access to information. However, at this time, no single wireless network is able to offer a wide-area data service with low latency, high bandwidth, and multiple users [1]. Mobile users in the most recent generation of wireless networks can switch between networks as needed and have access to the best connections that are appropriate for their service needs [2]. Location and handoff management are the two parts of mobility management. The system can track mobile users' locations through location management between communications. The process by which users sustain their

networks when they shift from one base station (BS) to another is known as handoff management [3]. Each cell in a wireless network is usually connected by a base station or access point. Mobile devices like smartphones, laptops, and IoT devices must switch from one cell to another as they move throughout the coverage area. Dropped calls, halted data transfers, and a frustrating user experience can occur if this isn't done properly [4]. In the field of mobility management for wireless networks, quick and seamless handoff is a key objective. The main objective of seamless handoff management is to make sure that this transition takes place without interruption or a drop in service quality [5].

Over the years, many schemes and methods have been created with the intention of enhancing wireless network performance, and a great deal of research has been done on wireless network congestion control [6]. The Transmission Control Protocol (TCP) is a widely used protocol on the modern Internet. The protocol establishes a connection between the ends of transmission and reception to support dependable data transport [7]. When transmitting a packet to the receiver, the transmitter initiates a timeout mechanism. In order to establish the proper timeout period, the transmitter continuously monitors the round-trip times (RTTs) for its packets [8]. Each packet that is received by the receiver is either implicitly or explicitly acknowledged by the transmitter. If the transmitter doesn't get an acknowledgement when the associated timeout period has passed, the packet is considered lost and is open to retransmission. The protocol uses a congestion window whose size is dynamically adjusted to control traffic flow from the transmitter to the receiver [9]. Consecutive packet losses are additionally introduced by temporary link disconnect during handoff [10]. Due to packet loss that is unrelated to congestion and varying round trip times, the TCP protocol may perform worse over wireless networks. Failures are indicated by packet loss or the reception of out-of-order packets. To avoid these failures, TCP employs flow control and congestion control methods based on the sliding window [11].

Seamless handoff management is essential for maintaining a consistent and uninterrupted user experience as devices move within a wireless network. Whether it's a user walking through a building while on a video call or a vehicle travelling through a cellular A, a consistent and uninterrupted user experience is required as devices move around a wireless network, which calls for seamless handoff management. The capacity to keep a steady connection is essential whether a user is on a video call while moving through a building, a car is moving through a cellular network's coverage area, or a smartphone switches between Wi-Fi and cellular connections [12]. Traditional TCP mechanisms find it difficult to handle the particular difficulties presented by wireless networks, such as fluctuating signal strengths, interference, and mobility-related problems. The Transmission Control Protocol (TCP)-PRN extension was created to improve the efficiency and reliability of data transmission over unreliable networks. This is accomplished by adding a partially reliable mechanism that enables applications to specify the significance of specific data packets [13]. In order to address the difficulties of handoff management in wireless networks, deep learning techniques, in particular recurrent neural networks (RNNs) and convolutional neural networks (CNNs), have shown great promise. These neural networks are able to continuously adapt to changing network conditions, learn patterns from historical data, and make intelligent choices [14]. Additionally, intelligent agents can learn the best handoff procedures by interacting with the network environment using reinforcement learning (RL)

techniques. In order to continuously improve handoff management over time, RL agents can modify their strategies based on observed performance [15].

The main contributions of this work are as follows:

To improve handoff prediction accuracy, the use of deep learning methods, such as CNN and SAEs, is one of the work's major contributions.

To enable senders to adapt the behaviour based on actual network congestion and predicted handoff events, DQN makes reinforcement learning possible.

TCP-PRN's congestion control decisions are made more effectively as a result of the incorporation of Tuna Swarm with Whale Optimization hybrid optimization techniques, namely the Tuna customized whale optimization algorithm.

The paper is organised as follows: Section 2 explains the related works of existing methods. The proposed methodology is described in section 3. The outcomes and analysis are done in the Section 4. Section 5 ends with the conclusion.

Literature Review

"Some of the recent research works related to handoff management were reviewed in this section."

In 2021, Wang S. et al. [16] suggested a multi-objective optimization paradigm-based optimized algorithm to address the incomplete evaluation of user and network influences throughout the handoff process. For each base station (BS), a Markov chain model was created in order to determine a more precise rate for the network state that reflected network performance. Then, they created a multi-objective optimization model in order to maximize the parameters of the state of the network and the user data-acquiring rate. The final VHO strategy was created by the researchers using the multi-objective genetic algorithm NSGA-II.

In 2019, Han Z. et al. [17] suggested DCRQN, a modern handoff management system based on deep RL, particularly DQN. The suggested method allowed networks to adapt their learning in time-varying dense WLANs by starting from scratch and learning from the behaviours and network status of actual users. The Markov decision process (MDP) was used to model the handoff decision because of the temporal correlation property. The suggested scheme in the modelled MDP was dependent on the current network statistics at the time of decisions. Additionally, fine-grained discriminative features were extracted using both the recurrent neural network and the convolutional neural network.

In 2022, Jha, K. et al. [18] created a new hybrid cuckoo search (CS) and genetic algorithm (GA) that is designed to improve throughput while reducing latency and the likelihood of handover failure in heterogeneous wireless systems. For the purpose of illustrating the benefit of their hybrid model, they concentrated on an optimized simulation framework. Throughput was increased by 17% and 8% using

the proposed hybrid technique compared to using cuckoo search and genetic algorithms separately, according to simulation analysis.

In 2023, Patil, M.B. and Patil, R. [19] aimed to make VHO more reliable and efficient for the HWN by enhancing the basic operating principle of earlier techniques. They had initially modelled the handover triggering methods to determine the proper location to start the handover based on the calculated coverage area of cellular base stations or WLAN access points. To find the best network for performing handover, they first eliminated any ineffective networks. As a result, they developed a network selection method based on a support vector neural network (Fractional-DE-based SVNN) that uses dolphin echolocation. Dolphin echolocation (DE) and fractional calculus (FC) were combined to create the Fractional-DE, which modified the DE algorithm's update rule based on the locations of the solutions in previous iterations.

In 2023, Khan, M.W. et al. [20] established a handoff method for optimum decision-making during the handoff. When more than one contradictory parameter was present, decisions were made using Multi-Criteria Decision Making (MCDM) methods. The ranking of the available networks was determined using the fuzzy technique for order performance by similarity to the ideal solution (TOPSIS) and the fuzzy analytical hierarchical process (FAHP) for weight calculation. The management of handoffs' overall QoS is optimized by incorporating parametric weights under various traffic classes, which also decreases the number of unnecessary handoffs.

In 2021, Patil, M.B. and Patil, R. [21] have developed an innovative method to enhance the energy effectiveness of various heterogeneous networks. The fractional squirrel-dolphin echolocation (FrSqDE) algorithm was created by merging the fractional dolphin echolocation (fractional DE) algorithm and the squirrel search optimization (SSA). The deep belief networks (DBNs) had been trained to choose the best weights using the suggested FrSqDE optimization algorithm. In this method, the suggested FrSqDE-based DBN received the network selection variables as input and used them to decide how best to handle vertical handoff.

In 2021, Bahra, N. and Pierre, S. [22] suggested a hybrid user mobility forecast approach to handover management in mobile networks. Utilizing a mobility model built on statistical models and DL algorithms, they first extracted user mobility patterns. To forecast a user's future trajectory, they used a gated recurrent unit (GRU) and a vector autoregression (VAR) model. Using the obtained prediction results, they then decreased the number of pointless handoff signalling messages and improved the handover process. To carry out their experiments, they used mobility data collected from actual users.

In 2021, Naresh M. et al. [23] introduced a method for performing Vertical Handover (VH) that combines Deep Residual Neural (DRN) and Wind-Driven Water Wave Optimization (WDWWO). Mobile users had a difficult time predicting Received Signal Strength (RSS) accurately when switching between networks. They used DRN with weight optimization based on WDWWO to address this issue, giving it the name Optimised DRN (ODRN). The proposed work had been put into practice on the NS2

platform, and the outcomes, like energy usage, had been assessed.

In 2019, Algeri, N. and Boukerche, A. [24] suggested a two-tier system based on machine learning for handover control in vehicular networks. The receiving signals from the Access Points were predicted using a recurrent neural network model, and the handover trigger decisions were then derived from those signals. A stochastic Markov model was also employed to choose the following AP. The network simulator NS-2 and SUMO mobility were used to assess the performance of their protocol.

In 2023, Han J. et al. [25] established the EdAR design, a novel trainable model with adaptable redundant packet scheduling that addresses the handoff problem in wireless mobile networks. EdAR is an experience-driven multipath scheduler. In various network environments, a skilled EdAR scheduler could react swiftly to altering network conditions and make precise scheduling decisions.

Proposed Methodology

The proposed methodology in this work aims to address the issues of performance degradation during handoffs in wireless networks. In order to improve handoff prediction and adaptive congestion control, an improved TCP-PRN (Path Recovery Notification) mechanism is proposed that makes use of DL techniques like Convolutional Neural Networks (CNNs), Stacked Autoencoders (SAEs), and reinforcement learning via Deep Q-Networks (DQN), in addition to hybrid optimization strategies. This improved TCP-PRN approach modifies the sender and receiver operations to support these innovative techniques, ultimately improving network performance and reliability during handoffs. Figure 1. shows the overall flow for the proposed work.

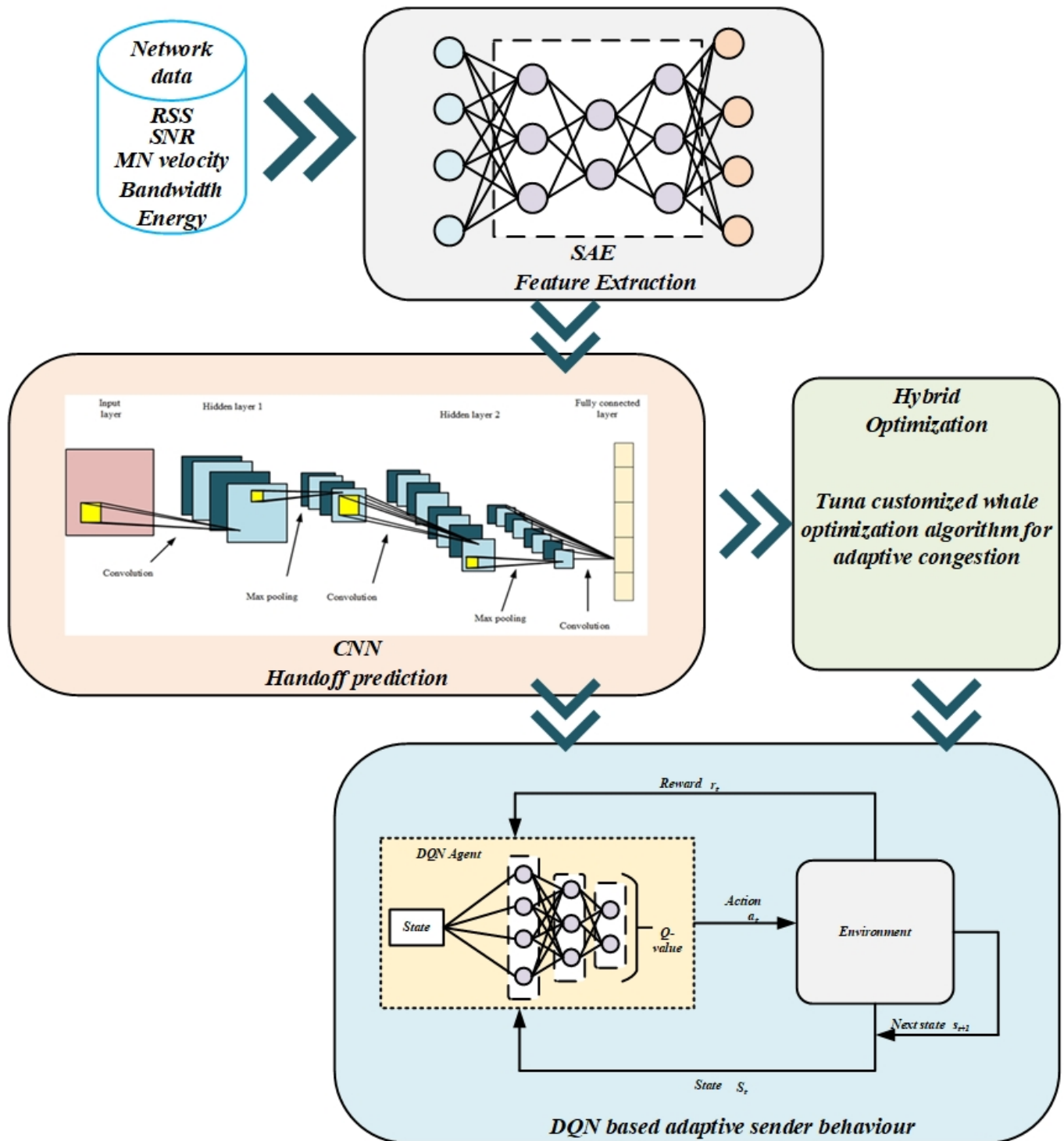


Figure 1. The overall proposed work

3.1 Network model

In this model, small cells with cellular coverage area are taken into consideration. Assume that B represents the base station (BS) with the formula and the set of mobile nodes (MN). Monitoring signal strength is a crucial part of the handoff procedure in wireless communication networks. It makes sure

the MN keeps an effective and dependable connection with the BS and starts a handoff when it's required. The wireless signal strength that the mobile node is currently receiving from its base station is measured and monitored by the node in real time. The network or system parameters define the threshold (T), which is a predetermined signal strength level. It serves as a benchmark for assessing whether the current connection is trustworthy and appropriate for ongoing communication. Typically, the threshold is set at a level that guarantees good signal quality while also being alert enough to notice when the signal quality starts to deteriorate. The network model is shown in the figure 2.

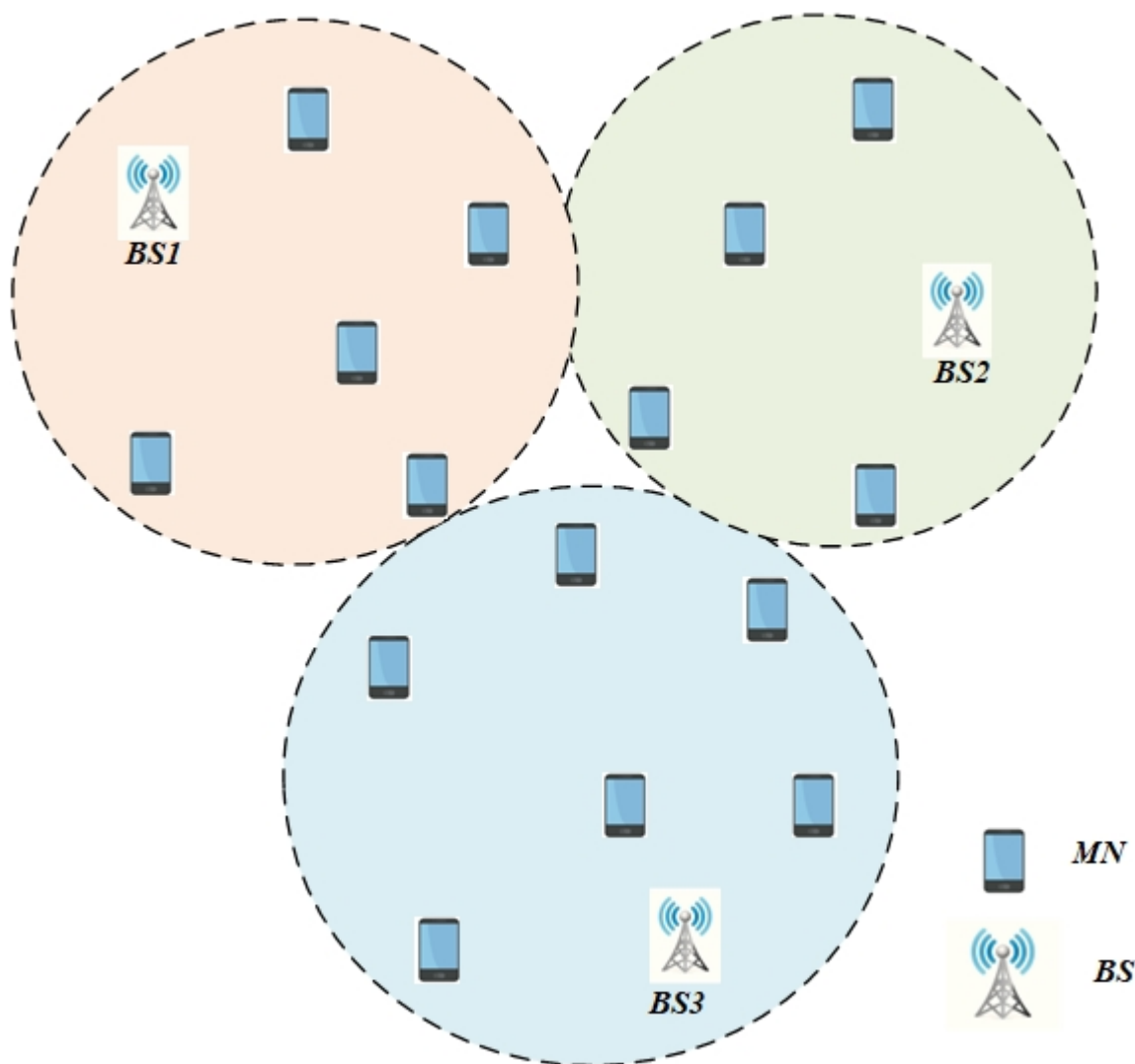


Figure 2. Network model

3.2 Network data features

The data features from the network model are as follows

Received Signal strength

The crucial network selection parameter, RSS, is initialized to the value of the current network under the threshold. The RSS indicates the link quality and signal strength. The smallest SNR is required for RSS detection, which stands for path loss due to the propagation environment, stands for the received signal strength, and the transmitted power from the signal.

Signal to Noise Ratio (SNR)

The SNR evaluates the desired signal strength in relation to background noise strength. SNR is described as the difference between the power of the network noise and the energy received from the signal.

Here is the strength of the network noise, which is the energy signal received per bit.

MN velocity

The choice of network may vary depending on MN speed. The handover process will be impacted by a higher velocity, which raises the likelihood of a failed handover.

Bandwidth

Bits are transmitted over networks at a rate of one per second. The corresponding procedure becomes delay-sensitive if the network data transmission bandwidth is sufficient. Higher bandwidth is needed for the handover process based on the likelihood of calls being blocked and dropped. While MN is moving through the network, bandwidth management is challenging. The network's link capacity is its bandwidth. Equation (4) performs bandwidth allocation.

The current user count is denoted by level I , and it denotes the level of user bandwidth allocation.

Energy consumption

For wireless devices, energy consumption must be limited. A new network with greater flexibility and lower energy consumption was introduced. Their battery life will be prolonged if the energy level drops. The closest BS uses less energy than BS, which is farther away from MNs. The mobile terminal must transfer to the closest BS with RSS above the threshold if the energy level is lower. More energy is consumed in the closest BS as the number of mobile users rises.

Here, stands for the path loss constant, stands for the travel distance, is the standard deviation, and is the exponent of path loss.

3.3 Feature extraction

The feature extraction and representation learning from complex data are particularly well-suited for Stacked Autoencoders (SAEs), a type of artificial neural network. SAEs are used to extract relevant features from network data, like signal quality, mobility, and strength. Hierarchical representations of input data are designed for SAEs to learn. This implies that they are able to capture features at various levels of abstraction. In order to process network data, SAEs first learn basic, low-level features before constructing increasingly complex, high-level representations. Since it enables the model to identify patterns and relationships within the data at various levels of granularity, this hierarchical feature extraction process is essential for improving the accuracy of handoff predictions. The architecture of SAE is shown in the figure 3.

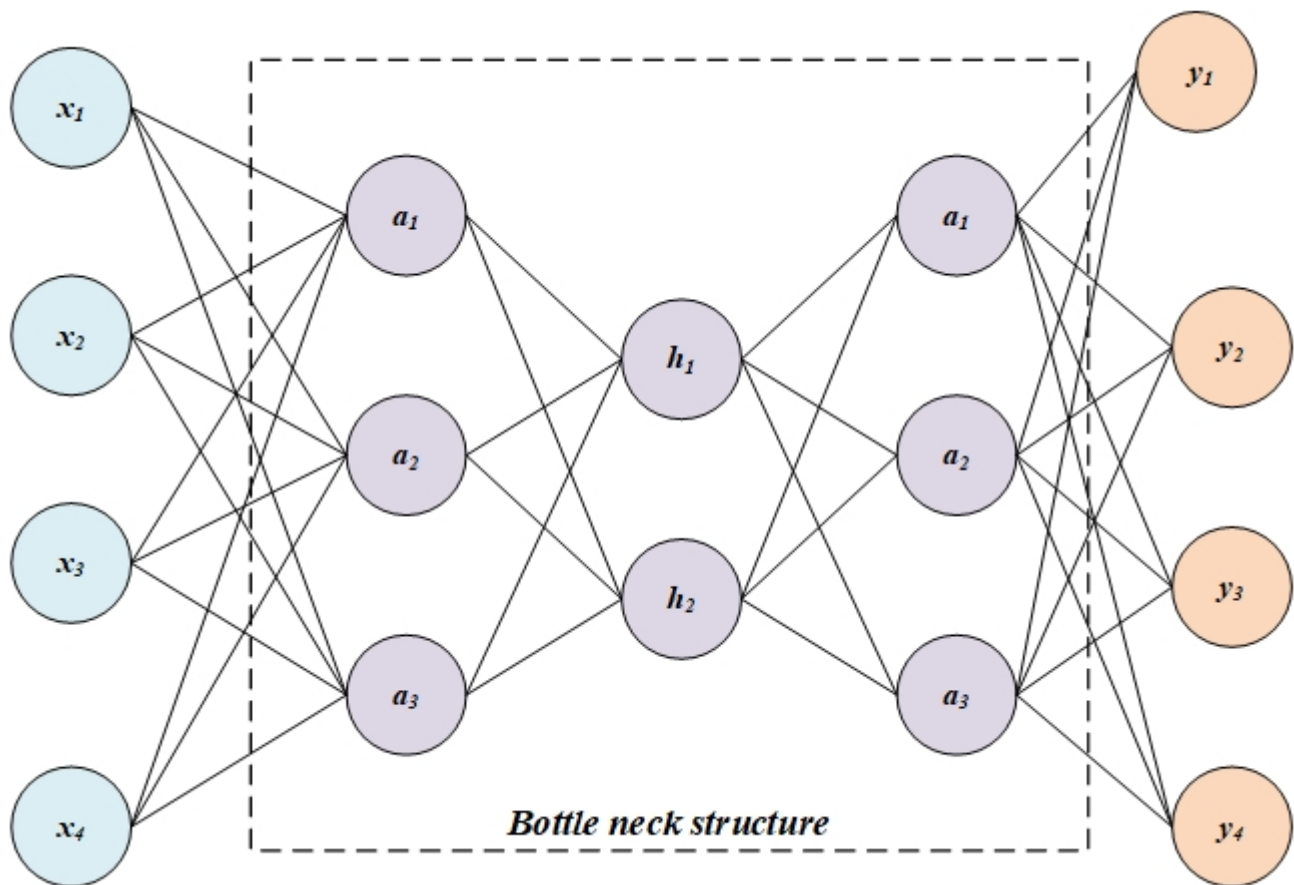


Figure 3. SAE architecture

3.3.1 Stacked Autoencoder

The autoencoder (AE), a feature compression algorithm accomplished by neural networks, is frequently used. Data-related decompression and compression are lossy. The only elements of the simplest AE structure are the input, hidden, and the output layer. An encoder, a decoder, and the selection of a loss function must be built in three easy steps to create an autoencoder. Most encoding

stages use feature reduction to map high-dimension data into low-dimension space. This task is carried out by a bottleneck framework in the autoencoder.

Reconstructing the data through the decoding process enables feedback with the loss function for the neural network training. The reverse aspect of the encoding process is typically thought of as decoding.

The loss function is a crucial element of the gradient descent method. The loss will be transmitted back to the hidden layer and updated there, changing the biases and weights of the neuron units.

An SAE is formed by assembling numerous autoencoders to increase AE's functionality. The advantage of the stacked self-encoder is that the feature extraction process becomes deeper as the number of layers increases.

3.4 Handoff prediction

In wireless communication, handoff prediction is a crucial task that aims to enhance the smooth transition of mobile devices from one base station (or access point) to another as they move through the network. When a mobile device moves, the process of handoff or handover is used to transfer an active call or data session to a new cell or access point. It is essential for preserving service quality and guarantees continuous connectivity. Handoff prediction is the process of determining, based on various network and device parameters, when a handoff is likely to be necessary. A smooth transition is ensured due to this prediction, which enables the network to get ready for the handoff beforehand. The quality of service for mobile users can be improved overall by reducing call drops, improving network performance, and accurately predicting handoffs. Due to their capacity to recognize spatial patterns in data, CNN, a subset of deep learning models, can be used successfully for this task. Handover prediction is the issue of foreseeing the connection status of a mobile device connected to BS. The features extracted from the SAE are then continued with CNN to predict the handoff.

3.4.1 CNN

CNN are a subclass of deep learning models that are especially effective at tasks involving spatial data, like image analysis. CNNs can be employed to classify spatial patterns in network-related data that may point to the need for a handoff in the framework of handoff prediction. These patterns may be influenced by variables such as mobile device location, signal strength, and signal quality.

(i) Convolution layer

The convolution layer (CL) applies a kernel or filter to the assigned signal and convolves over it. A filter is a collection of connections that are frequently used across the entire input. The feature map that this filter generates is the output layer. The neuron is activated if the filter assisting a neuron's

action recognizes a suitable position at the layer before. This model employed the ReLU (Rectified Linear Unit) activation function.

(ii) Activation function

Any neural network performs better when the input is somewhat nonlinear. Each output of the convolution layer receives the activation function. The output of the convolution layer uses ReLU activation units, which are optimized by the new hybrid method HILCA, which is also used to store the output.

(iii) Pooling layer

To address the issue of local translation invariance in the pooling layer, the input feature maps are condensed. A list of the features on the input feature map that are available in samples is useful. Due to the pooling operation, the feature map interpretation is insensitive to even slight changes in the data. It also aids in avoiding the issue of overfitting. The two types of pooling are average pooling and maximum pooling. The max-pooling operation computes the greatest value, while the average pooling operation computes the average value for each segment of the feature map.

Fully connected layer (FCL)

The FCL of the CNN model is an essential element. The CNN model may also incorporate multiple FCLs. The most practical weights are chosen by the FCL using backpropagation. The neurons in the FCL are weighted appropriately to highlight the most significant weights. Then, the neurons in the FCL vote on each label, and the classification's final result is decided by the winner. The SoftMax activation function is typically used for multi-class classification tasks. Figure 4 depicts the structure of CNN.

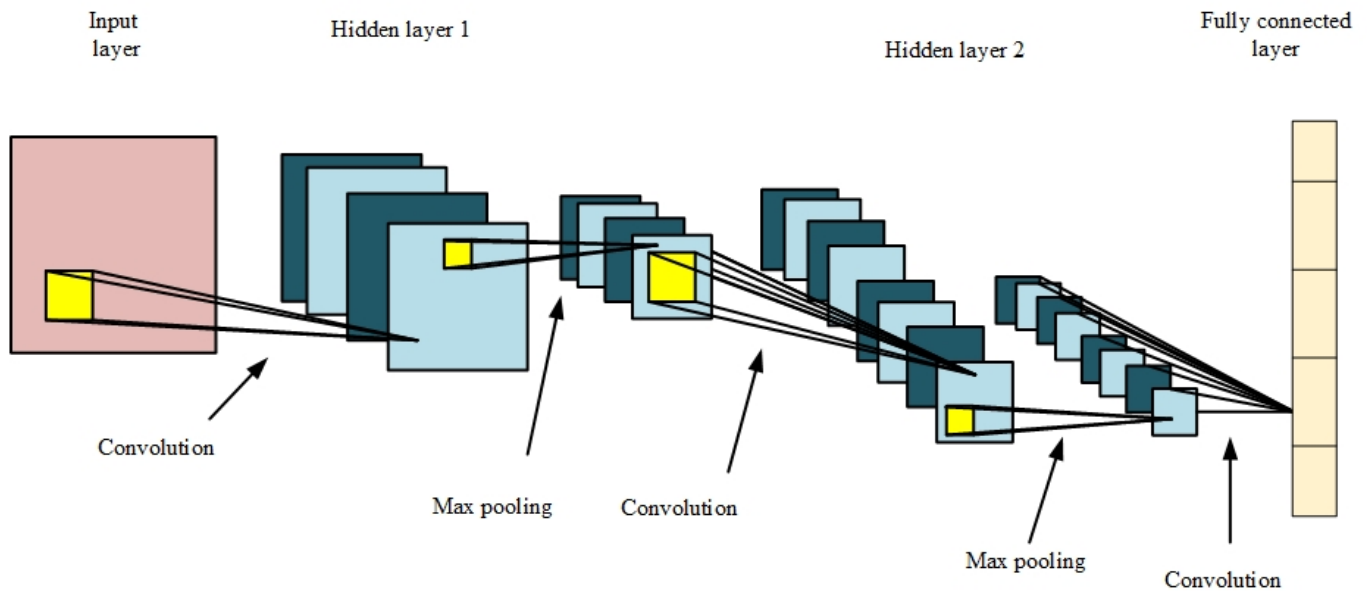


Figure 4. CNN architecture

3.5 TCP-PRN

The fundamental idea behind TCP-PRN is to quickly recapture from the negative effects of a handoff-induced disconnection of the temporal link. The impact of packet losses during the disconnect is reduced when a TCP receiver on an MN informs its TCP sender on an FN of the temporal disconnection using a special ack. The TCP recipient then resends the lost packets and recovers the reduced cwnd and ssthresh if a false RTO occurs. Both the transmitter and the receiver must modify their TCP-SACK handling in order to support TCP-PRN.

PRN option:

A wireless link may have briefly been disconnected and then reconnected using this option, which helps to inform the sender. One of the three condition flags—reconnection (RC), partial loss (PL), or no packet arrival (NPA)—is selected when the option is selected.

ocwnd

This parameter is a duplicate of cwnd. If an RTO occurs as a result of a disconnect, it is used to restore cwnd.

ossthresh

This parameter is a duplicate of ssthresh. If an RTO happens as a result of a disconnect, it is used to

restore ssthresh.

lsent_time:

This variable specifies when the TCP sender last sent a packet during a window as a result of an RTO or a fast retransmit.

PRN timer:

The PRN option with NPA is sent to the sender in the event that the timer expires, alerting them that there may have been a loss of all packets.

The negative effects of the handoff-induced disconnection of the temporal link are to be recovered by PRN. A special acknowledgement is sent by a TCP receiver when it is connected to a new access point following a disconnection or handoff. This is composed of two parts: a TCP SACK and the PRN option. The PRN option is important in informing the sender that a wireless link was briefly cut off and then reconnected. Three condition flags are selected as the option's setting. Reconnection (RC), partial loss (PL), and no packet arrival (NPA) are the three. The sequence numbers of the lost packets are listed in the TCP SACK. This acknowledgement informs a TCP sender of the packets that were lost as a result of the temporal link breaking. The receiver will send an acknowledgement with a PRN option set to the condition flag once it has determined that the wireless link has been restored. If a new packet arrives at the receiver prior to the PRN timer expiration, there may be gaps in the sequence values. The receiver then transmits an acknowledgement packet with a PRN option set to the PL flag. This makes sure that only the lost packets are transmitted by the sender. The receiver then stops the PRN timer. When the PRN timer expires, and no new packets reach the receiver, the receiver transmits a packet with acknowledgement with just the PRN option set to NPA. This forces the sender to carry out the process by sending all unacknowledged packets. Packet loss is divided into two categories: congestion loss and link loss using the Enhanced Path Recovery Notification (EPRN) protocol. Packets are sent from the sender to the receiver. The packet received is acknowledged by the receiver. The sender is aware that a packet has been lost thanks to the Duplicate Acknowledgement (DACK) message that the receiver sent. Either congestion or a link error could be to blame for the packet loss. The algorithm categorizes the packet loss appropriately after receiving the DACK. If the loss is determined to be due to a link failure, PRN is immediately invoked to resend the lost packets. The TCP slow start algorithm is used to retransmit the lost packets if the loss is caused by congestion. The enhanced PRN is shown in the figure 5.

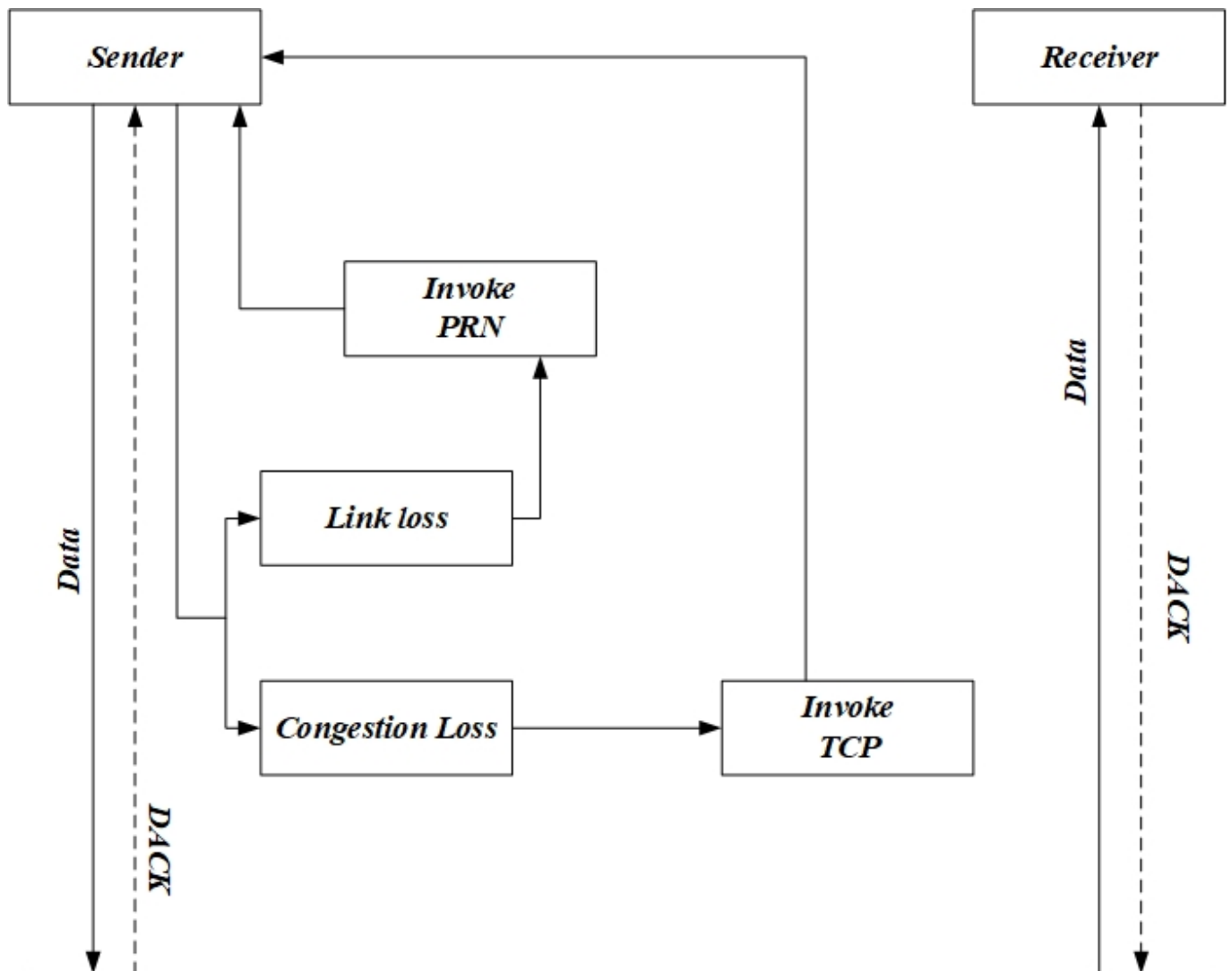


Figure 5. Enhanced PRN

3.6 Adaptive congestion control

The purpose of combining TSO and WOA for congestion control is to take advantage of the strengths that these two nature-inspired algorithms have in addition to one another. While WOA's spiral mechanisms, inspired by humpback whales, can help strike a balance between exploration and exploitation, TSO's cooperative and coordinated search behaviour, inspired by tuna swarms, can aid in effectively exploring the search space. The hybrid algorithm may be able to find superior solutions to congestion control issues more successfully by combining these two approaches. The TCP-PRN (Path Recovery Notification) congestion control decisions are significantly improved by the Tuna customized whale optimization. These strategies take advantage of the advantages of both optimization techniques to dynamically modify congestion control parameters and experiment with different parameter settings. Based on the behaviour of the network and the requirements of the particular mechanism, the congestion control parameters *cwnd* in TCP-PRN are optimized using hybrid

optimization. The objective is to accommodate changing network conditions while avoiding congestion, maximizing resource utilization, and maintaining effective data transmission.

3.6.1 Tuna Customized Whale Optimization

The tuna swarm optimization algorithm creates the initial swarm in the search space at random during the swarm initialization phase.

Spiral foraging strategy with spiral updating position

Search agents at different distances and the prey use this approach when it is first calculated. It controls the updating rate (α) at which the spiral trajectory is formed.

Sardines, herring, and other small schooling fish constantly change their swimming directions when they come into contact with predators, forming a dense formation that makes it difficult for the predator to lock on to a target. At this stage, the tuna school forms a tight spiral formation to pursue its prey. Even though most fish in the community have a poor sense of direction, when a small group swims gradually in one direction, the nearby fish slowly change their path until they eventually come together in a large group with the same goal and start to hunt. Schools of tuna communicate with one another as well as spiral their prey. Since every tuna follows the one before it, nearby tuna can communicate with one another and exchange information.

The reference point chosen at random from the tuna swarm is. The trend weight coefficient directs the tuna to swim to the ideal individual or a set of adjacent individuals at random. The trend weight coefficient determines how closely a tuna swims to a fish in front of it. The parameter that determines how far apart the tuna individual is from either the ideal individual or a randomly chosen reference individual is called the distance parameter.

Parabolic Foraging Strategy

The agents constantly alter their swimming direction when they come across predators, taking advantage of their advantage in speed. Predators have a very difficult time catching them. The prey will serve as a stopping point for the tuna swarm as it pursues more prey. Each tuna follows the one before it during predation, and the entire tuna swarm shapes a parabola to encircle the prey. The mathematical model of the tuna swarm's parabolic foraging is as follows, assuming that the probability of either strategy being chosen by the swarm is 50%:

3.7 Adaptive Sender Behaviour

An innovative idea known as adaptive sender behaviour using Deep Networks (DQN) involves

teaching an artificial intelligence agent to decide how to modify the behaviour of the sender in a dynamic network environment. Adaptive sender behaviour is critical, as network conditions can change quickly due to factors like user mobility and network congestion. To maintain high-quality communication while effectively using network resources, the sender's actions, such as adjusting transmission rates or starting handoff procedures, are adjusted. A class of reinforcement learning algorithms called Deep Networks (DQNs) can learn to make the best decisions in changing environments. A DQN gains the ability to accept network state information, handoff predictions, and congestion data as input and produce adaptive sender actions. The DQN can decide on actions that will maximize long-term performance because it employs a Q-value function to estimate the expected cumulative reward for each potential action.

3.7.1 DQN

A quintuple of the letters is known as a Markov decision process. With DQN, state representation is a crucial part of adaptive sender behaviour. It contains information on the current network conditions, anticipated handoff events, and congestion. Because it directly affects the sender's capacity to make wise decisions, the state representation's design is essential. The set of potential actions that the sender may take in response to the observed state is defined by the action space. Actions that can be taken in the surroundings of adaptive sender behaviour include Changing the transmission rate to boost or reduce data throughput and switching to a new base station or access point by starting the handoff procedures.

Reinforcement learning's central concepts are policy optimization and a trial-and-error mechanism. Agents tweak their policies by experimenting with various options in order to identify the best course of action that will benefit the environment the most. The agent will always select the action that can yield the highest Q value in each state when Q-learning is used to generate the action, which is called the greedy policy. The function is the central concept of Q-learning. After selecting an action in the state, the function signifies the cumulative reward up until the current episode ends.

To maximize expected cumulative rewards, the sender should act in a way that maximizes the optimal Q-values that are learned during training. The oscillation issue is resolved by DQN during training using dual neural networks and experience replay. In contrast to Q-learning, where the parameters are revised right away after acquiring experience in each step, DQN saves the experience gained in each step to an experienced pool, which is dignified. To update the parameters, DQN will select a random sample of experience from the experience pool. The states in the DQN were employed as the inputs of a DNN to produce Q-values that matched each action. DQN, which can address the dimensionality issue associated with sizable state spaces, incorporates DNN and Q-learning.

The state was used as a DNN input at time slot t , and the outputs for each action were the Q-values. For the various actions, there were a total of M outputs. The DQN agent was trained to update the Q-

values by calculating the value of the Q-table, which resulted in an update. The agent could acquire the Q-values after convergence by entering the states into the DQN. The DQN trained the RL processes by using experience replay to address the problems of information association and non-static distribution. Earlier learning experiences were stored in the experience buffer E of the DQN algorithm with a predetermined capacity CE.

The sender can proactively modify its transmission parameters if a handoff is anticipated to guarantee that packets are retrieved on the new path. The sender can dynamically lower its data rate to avoid network congestion if congestion is detected. This strategy enables the sender to make wise choices that enhance the performance and dependability of the network. The DQN model is shown in the figure 6.

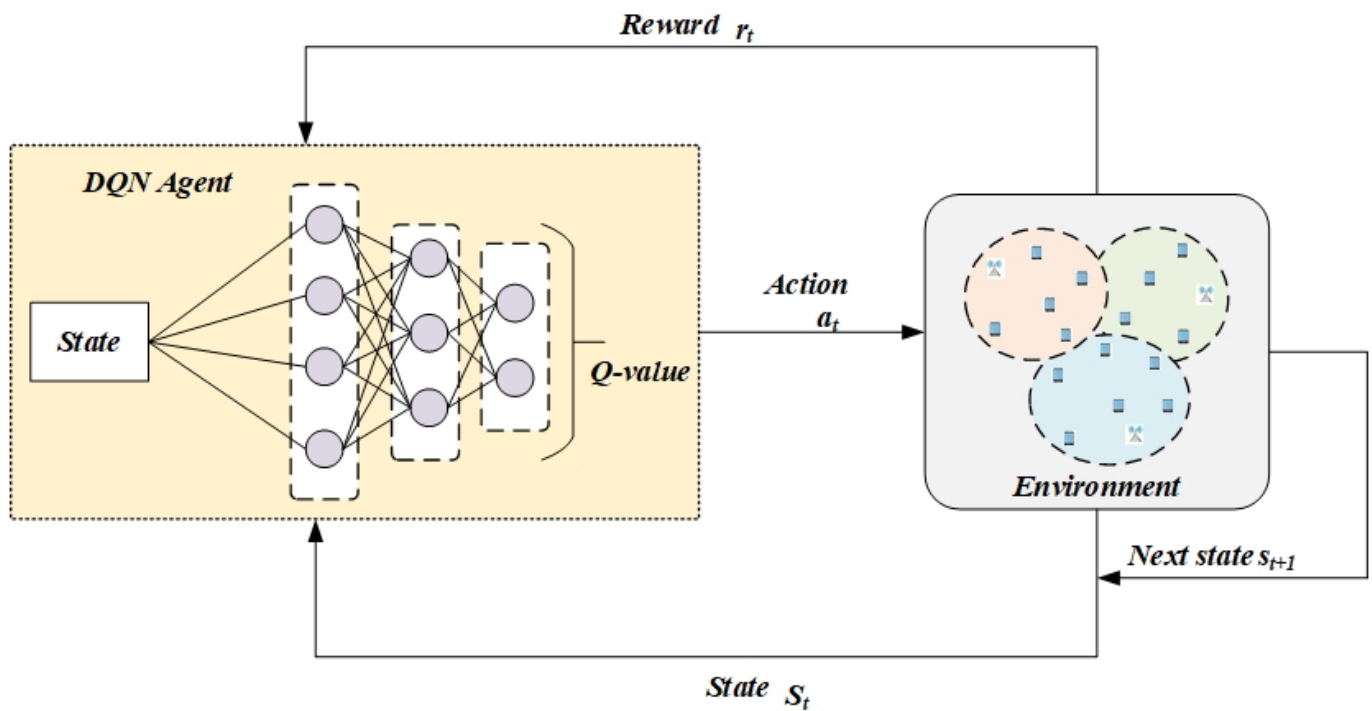


Figure 6. DQN

Result and Discussion

The proposed network is implemented in the NS2 network simulator, and the proposed method is compared with the existing techniques like NSGA-II [16], Fractional squirrel-dolphin echolocation (FrSqDE) [21] and Wind-Driven Water Wave Optimization (WDWWO) [23]. The proposed method is compared for the 100 nodes for the different metrics like Delay, delivery ratio, energy, network lifetime and throughput.

4.1 Performance metrics

The simulation is conducted using a variety of performance metrics, which are described as follows:

Throughput

The throughput represents all data rates sent across the network at a particular moment.

Delay

The delay parameter shows the total amount of time required to transmit data, regardless of attacks.

Delivery ratio

The packet delivery ratio (PDR), which is a network metric, is the proportion of total packets delivered to total packets sent from source nodes to destination nodes.

Network lifetime

The period of time between deployment and the point at which the network stops operating is known as the network lifetime.

4.2 Performance analysis

In this section, the performance is analysed graphically for the evaluation metrics. In Table 1, four different optimization algorithms—NSGA-II, FrSqDE, WDWWO, and the proposed algorithm—are compared in terms of delay times across various numbers of nodes.

Table 1. Delay in sec for the proposed and the existing

No of nodes	NSGA-II	FrSqDE	WDWWO	(TCWO) Proposed
25	4.9	4.1	3.9	2.1
50	15	22	23.1	9.2
75	24	37.1	38	11.4

100	26	23	24.2	13.8
-----	----	----	------	------

The proposed TCWO algorithm consistently performs better in terms of delay reduction than the other three algorithms. For instance, with 25 nodes, TCWO only experiences a delay of 2.1 seconds, compared to delays of 4.9, 4.1, and 3.9 seconds for NSGA-II, FrSqDE, and WDWWO, respectively. As the number of nodes rises, this pattern persists, with TCWO consistently showing the lowest delays when compared to the other algorithms. Figure 7 shows the graphical comparison of the delay.

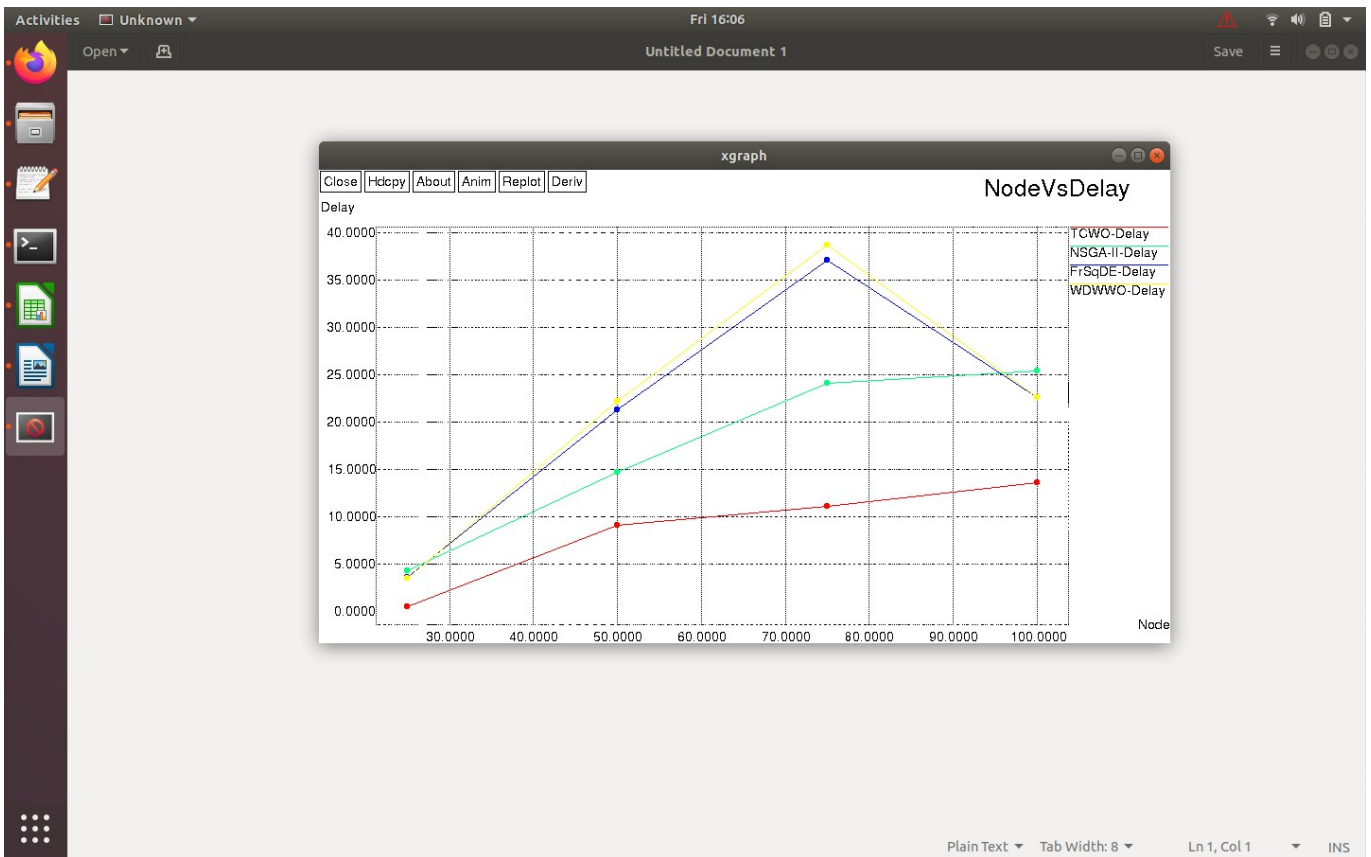


Figure 7. Node vs Delay analysis

Table 2 compares the delivery ratios for four different optimization algorithms, including the proposed algorithm (also known as TCWO), FrSqDE, WDWWO, and NSGA-II, across various node counts.

Table 2. Delivery ratio comparison for the proposed and the existing

No of nodes	NSGA-II	FrSqDE	WDWWO	(TCWO) Proposed
25	0.91	0.92	0.93	0.98

50	0.6	0.33	0.3	0.61
75	0.29	0.12	0.1	0.48
100	0.08	0.03	0.03	0.39

Across all possible node count scenarios, the proposed TCWO algorithm always exceeds the other three algorithms with regard to delivery ratio. For instance, TCWO achieves a remarkable delivery ratio of 0.98 when there are 25 nodes, showing that a substantial number of data packets are successfully delivered. In contrast, the delivery ratios for NSGA-II, FrSqDE, and WDWWO are lower, at 0.91, 0.92, and 0.93, correspondingly. The delivery ratio graphical analysis is depicted in the figure 8.

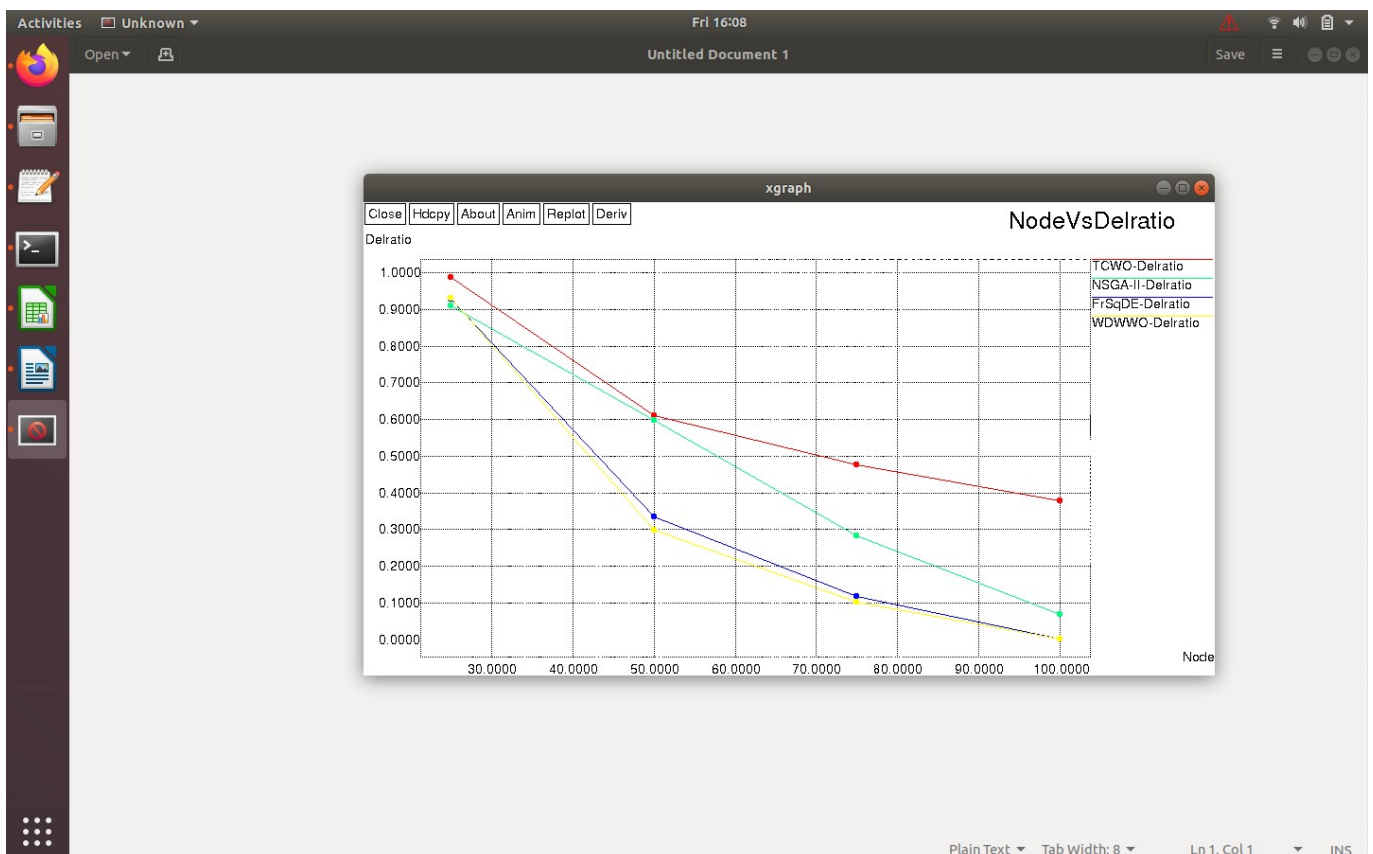


Figure 8. Node vs Delratio analysis

Table 3 compares the energy consumption in joules (J) for four different optimization algorithms, including NSGA-II, FrSqDE, WDWWO, and the proposed algorithm.

Table 3. Energy (J) comparison for the proposed and the existing

No of nodes	NSGA-II	FrSqDE	WDWWO	(TCWO) Proposed
25	150	122	112	98
50	92	62	70	60
75	89	58	60	48
100	68	49	47	38

In every node count, the proposed TCWO algorithm consistently outperforms the other three algorithms in terms of energy efficiency. With 25 nodes, for instance, TCWO uses the least amount of energy (98 J), while NSGA-II, FrSqDE, and WDWWO use 150, 122, and 112 J, correspondingly. Figure 9 illustrates the energy graphical comparison.

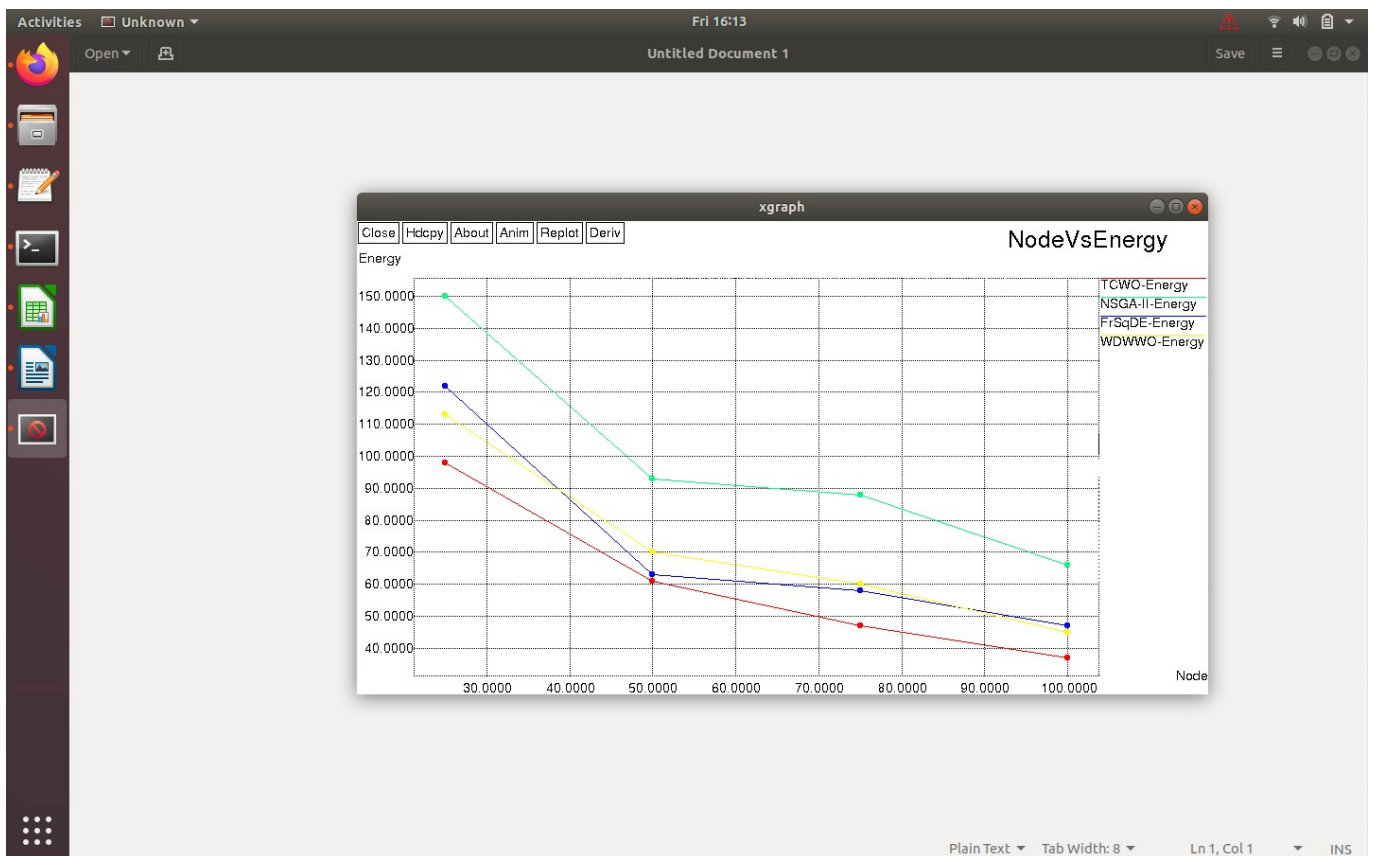


Figure 9. Node vs Energy analysis

Table 4 compares the network lifetime values for four different optimization algorithms, including the proposed algorithm, FrSqDE, WDWWO, and NSGA-II, across a variety of node counts.

Table 4. Network lifetime comparison for the proposed and the existing

No of nodes	NSGA-II	FrSqDE	WDWWO	(TCWO) Proposed
25	150	150	110	460
50	146	98	50	260
75	52	49	30	170
100	37	25	20	120

The proposed TCWO algorithm consistently surpasses the other three algorithms in terms of network lifetime for all node count scenarios. For instance, with 25 nodes, TCWO achieves a remarkable network lifetime of 460 units, far exceeding the network lifetimes of NSGA-II, FrSqDE, and WDWWO, which are correspondingly 150, 150, and 110 units. The network lifetime comparison analysis is shown graphically in Figure 10.

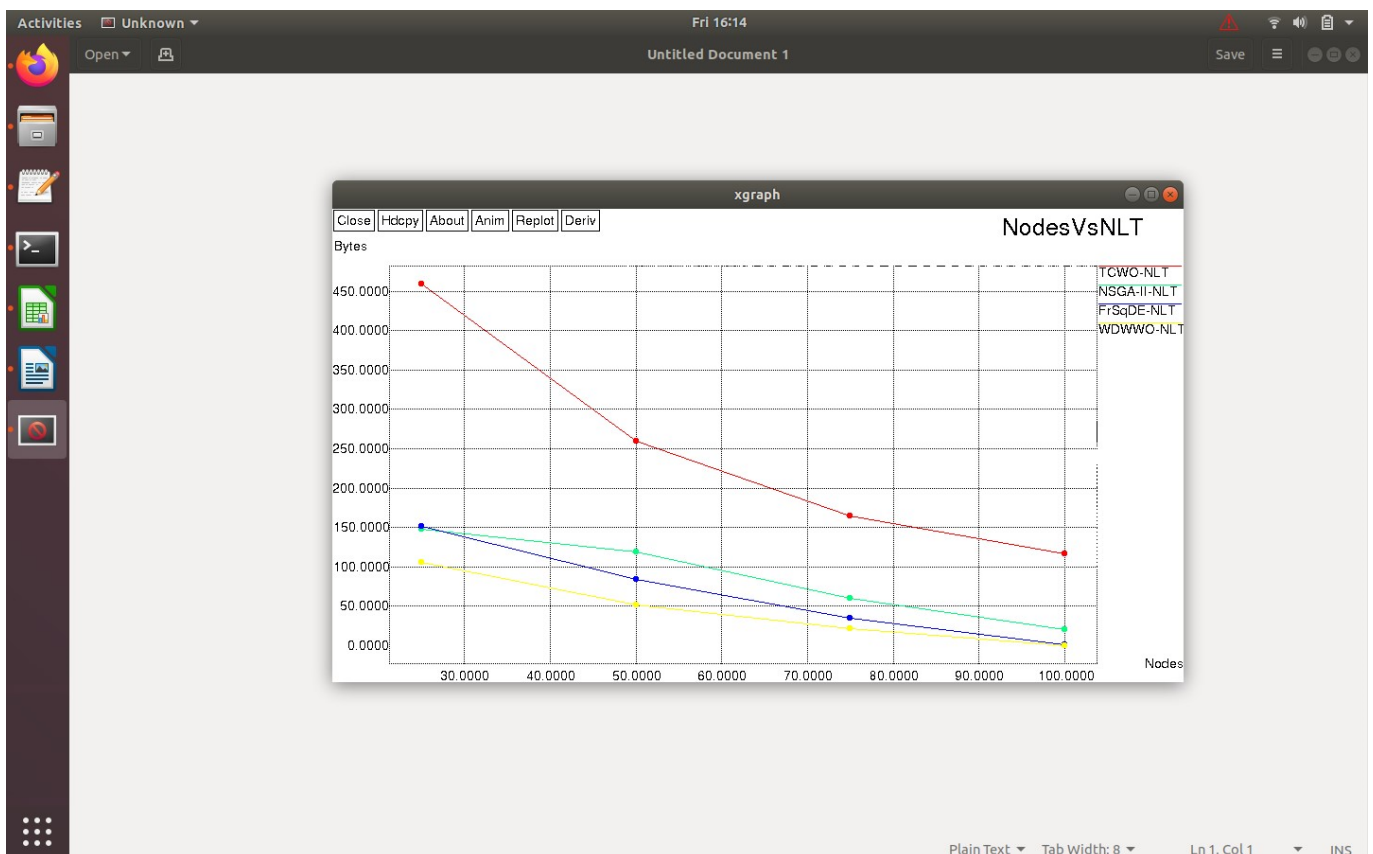


Figure 10. Node vs Network Lifetime Analysis

Table 5 represents the throughput comparison for the proposed and existing nodes 25, 50, 75, and 100.

Table 5. Throughput (kbps) comparison for the proposed and the existing

No of nodes	NSGA-II	FrSqDE	WDWWO	(TCWO) Proposed
25	380	500	530	780
50	300	298	280	430
75	150	120	110	290
100	80	30	31	200

In terms of throughput, the proposed TCWO algorithm consistently outperforms the other three algorithms for all node count scenarios. For instance, with 25 nodes, TCWO surpasses NSGA-II, FrSqDE, and WDWWO, which each achieve throughputs of 380, 500, and 530 kbps, to achieve the highest throughput at 780 kbps. Figure 11 depicts the throughput comparison graphical analysis.

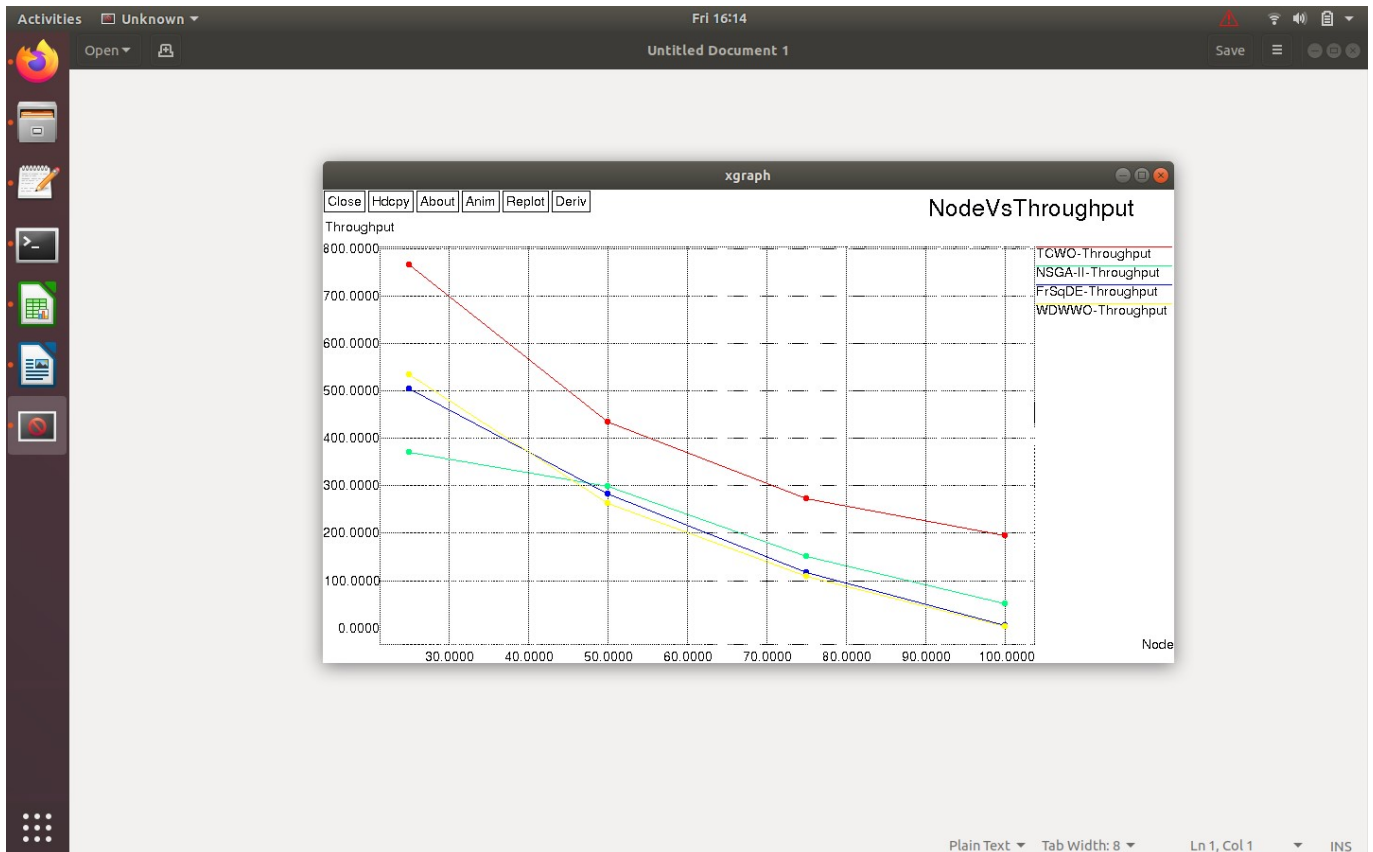


Figure 11. Node vs Throughput analysis

Conclusion

An improved TCP-PRN mechanism was developed in this study to help wireless networks cope with handoff events and brief link disconnections. Although TCP-PRN aimed to mitigate the effects of these problems, accurate handoff prediction and efficient management of link disconnections remained difficult in practical deployments. Adaptive congestion control and deep learning-based handoff prediction are the main contributions of this improved TCP-PRN mechanism. With the help of proactive packet recovery strategies, a CNN-based approach can predict handoff events with accuracy. SAEs help to extract features from network data, improving prediction precision. Senders can modify their behaviour based on anticipated handoff events and the state of the network's congestion due to the incorporation of DQN. Additionally, TCP-PRN's congestion control choices are enhanced by hybrid optimization TCWO. This strategy guarantees flexibility and effectiveness during handoff events in addition to improving network dependability. The robustness and adaptability of wireless network protocols are expected to be improved by the union of deep learning and hybrid optimization. The proposed TCWO algorithm consistently outperforms existing approaches like NSGA-II, Fractional squirrel-dolphin echolocation (FrSqDE), and Wind-Driven Water Wave Optimization (WDWWO) across a range of metrics, including delay, delivery ratio, energy consumption, network lifetime, and throughput. This indicates that the enhanced TCP-PRN mechanism holds great promise for the deployment of wireless networks in the real world, providing a comprehensive solution to deal with the problems posed by handoff events and link disconnections while enhancing network performance.

References

- Desai, A.M. and Jhaveri, R.H., 2019. Secure routing in mobile Ad hoc networks: A predictive approach. *International Journal of Information Technology*, 11(2), pp.345-356.
- Lu, H., Zhang, Y., Li, Y., Jiang, C. and Abbas, H., 2020. User-oriented virtual mobile network resource management for vehicle communications. *IEEE Transactions on Intelligent Transportation Systems*, 22(6), pp.3521-3532.
- Kumari, S. and Singh, B., 2019. Data-driven handover optimization in small cell networks. *Wireless Networks*, 25, pp.5001-5009.
- Leng, T., Xu, Y., Cui, G. and Wang, W., 2021. Caching-aware intelligent handover strategy for LEO satellite networks. *Remote Sensing*, 13(11), p.2230.
- Omheni, N., Bouabidi, I., Gharsallah, A., Zarai, F. and Obaidat, M.S., 2018. Smart mobility management in 5G heterogeneous networks. *Iet Networks*, 7(3), pp.119-128.

Qureshi, K.N., Abdullah, A.H., Kaiwartya, O., Iqbal, S., Butt, R.A. and Bashir, F., 2018. A dynamic congestion control scheme for safety applications in vehicular ad hoc networks. *Computers & Electrical Engineering*, 72, pp.774-788.

Kumar, V. and Tyagi, N., 2021. Device-centric data reordering and buffer management for mobile Internet using Multipath Transmission Control Protocol. *International Journal of Communication Systems*, 34(17), p.e4973.

Mishra, T.K., Sahoo, K.S., Bilal, M., Shah, S.C. and Mishra, M.K., 2023. Adaptive congestion control mechanism to enhance TCP performance in cooperative IoV. *IEEE Access*, 11, pp.9000-9013.

Li, W., Song, W., Lu, Q. and Yue, C., 2020. Reliable congestion control mechanism for safety applications in urban VANETs. *Ad Hoc Networks*, 98, p.102033.

Li, H., Wang, Y., Sun, R., Guo, S. and Wang, H., 2019, April. Delay-based congestion control for multipath TCP in heterogeneous wireless networks. In *2019 IEEE Wireless Communications and Networking Conference Workshop (WCNCW)* (pp. 1-6). IEEE.

Saedi, T. and El-Ocla, H., 2021. TCP CERL+: Revisiting TCP congestion control in wireless networks with random loss. *Wireless Networks*, 27, pp.423-440.

Yap, K.L., Chong, Y.W. and Liu, W., 2020. Enhanced handover mechanism using mobility prediction in wireless networks. *PloS one*, 15(1), p.e0227982.

Sinky, H., Hamdaoui, B. and Guizani, M., 2019. Seamless handoffs in wireless HetNets: Transport-layer challenges and multi-path TCP solutions with cross-layer awareness. *IEEE Network*, 33(2), pp.195-201.

Priyanka, A., Gauthamarayathirumal, P. and Chandrasekar, C., 2023. Machine learning algorithms in proactive decision making for handover management from 5G & beyond 5G. *Egyptian Informatics Journal*, 24(3), p.100389.

Mollet, M., Kaijage, S. and Michael, K., 2021. Deep reinforcement learning based handover management for millimeter wave communication.

Wang, S., Deng, H., Xiong, R., Liu, G., Liu, Y. and Liu, H., 2021. A multi-objective model-based vertical handoff algorithm for heterogeneous wireless networks. *EURASIP Journal on Wireless Communications and Networking*, 2021(1), pp.1-18.

Han, Z., Lei, T., Lu, Z., Wen, X., Zheng, W. and Guo, L., 2019. Artificial intelligence-based handoff management for dense WLANs: A deep reinforcement learning approach. *IEEE Access*, 7, pp.31688-31701.

Jha, K., Gupta, A., Alabdulatif, A., Tanwar, S., Safirescu, C.O. and Mihaltan, T.C., 2022. CSVAG: Optimizing Vertical Handoff Using Hybrid Cuckoo Search and Genetic Algorithm-Based

Approaches. Sustainability, 14(14), p.8547.

Patil, M.B. and Patil, R., 2023. A network controlled vertical handoff mechanism for heterogeneous wireless network using optimized support vector neural network. International Journal of Pervasive Computing and Communications, 19(1), pp.23-42.

Khan, M.W., Khan, U.S., Saleem, M.M. and Rashid, N., 2023. Multi-criteria Handoff Decision Making Algorithm for Seamless Mobility in Heterogeneous Wireless Networks. Journal of Communications, 18(3).

Patil, M.B. and Patil, R., 2021. Fractional squirrel-dolphin echolocation with deep belief network for network-controlled vertical handoff in disparate and heterogeneous wireless network. International Journal of Communication Systems, 34(12), p.e4893.

Bahra, N. and Pierre, S., 2021, May. A Hybrid User Mobility Prediction Approach for Handover Management in Mobile Networks. In Telecom (Vol. 2, No. 2, pp. 199-212). MDPI.

Naresh, M., Venkat Reddy, D. and Ramalinga Reddy, K., 2021. Vertical handover in heterogeneous networks using WDWWO algorithm with NN. International Journal of Electronics, 108(12), pp.2078-2099.

Aljeri, N. and Boukerche, A., 2019. A two-tier machine learning-based handover management scheme for intelligent vehicular networks. Ad Hoc Networks, 94, p.101930.

Han, J., Xue, K., Li, J., Zhuang, R., Li, R., Yu, R., Xue, G. and Sun, Q., 2023. EdAR: An Experience-driven Multipath Scheduler for Seamless Handoff in Mobile Networks. IEEE Transactions on Wireless Communications.