Intelligent Automation & Soft Computing
DOI: 10.32604/iasc.2023.031200
Article

Tech Science Press

Generative Adversarial Networks for Secure Data Transmission in Wireless Network

E. Jayabalan[*] and R. Pugazendi

Department of Computer Science, Government Arts College (A), Salem, 636007, Tamil Nadu, India
*Corresponding Author: E. Jayabalan. Email: ejksrcas@gmail.com

Received: 12 April 2022; Accepted: 08 June 2022

Abstract: In this paper, a communication model in cognitive radios is developed and uses machine learning to learn the dynamics of jamming attacks in cognitive radios. It is designed further to make their transmission decision that automatically adapts to the transmission dynamics to mitigate the launched jamming attacks. The generative adversarial learning neural network (GALNN) or generative dynamic neural network (GDNN) automatically learns with the synthesized training data (training) with a generator and discriminator type neural networks that encompass minimax game theory. The elimination of the jamming attack is carried out with the assistance of the defense strategies and with an increased detection rate in the generative adversarial network (GAN). The GDNN with game theory is designed to validate the channel condition with the cross entropy loss function and back-propagation algorithm, which improves the communication reliability in the network. The simulation is conducted in NS2.34 tool against several performance metrics to reduce the misdetection rate and false alarm rates. The results show that the GDNN obtains an increased rate of successful transmission by taking optimal actions to act as a defense mechanism to mislead the jammer, where the jammer makes high misclassification errors on transmission dynamics.

Keywords: Generative adversarial learning neural network; Jammer; Minimax game theory; Attacks

1 Introduction

One of the key assumptions of theoretical confidentiality of information is that users always have to send data. Users fail to provide data in wireless contexts such as cognitive radios. If traffic is explosive, theoretical information measurements, including capacity and secrecy, cannot measure the system performance. Jamming is a frequent denial of service attack wherein malicious nodes intend to disrupt continuous communication between legitimate nodes.

The data receiving at a node or base station (BS) varies dynamically, and if the jammer fail in knowing the status of the queue, the randomness of the data entry is used to improve system performance. In the study [1,2], a theoretical framework on games is examined to evaluate the effect of random data arrival in alleviating the jamming attack. In [3,4], a game theory examines the influence of jamming over a collusive channel. The research already under way does not examine the effect of cognitive radios in mitigating jamming attacks in the fading environment when data arrives at random. The influence of jamming on delaying cognitive radio performance when users have various antennas also needs to be understood. This also raises an interesting topic about how diversity in time and space might be used to improve system performance.

In several different scenarios [5,6], the influence of jamming on system functioning has been examined. An examination of linked jamming in theoretical information can be found in [7,8]. Multifunctional cases [9,10] including a multi-user access channel with associated jamming. In the presence of a jammer, the cognitive radio environment is examined, and a joint anti-jamming technique to reduce jamming attacks was developed. Game theory is utilized to investigate various jamming skills because of the nature on conflicting interests between the jammer and transmitter.

The capacity of the jammer to transfer power and information concerning at which frequency, the signal is sent which is a key to a successful attack. The reason for this is simple, because the jammer noise must be powerful enough to transmit on the same band with the reduction in signal noise ratio (SNR). The transmission between terrestrial terminals is made via a satellite relayed for satellite communications. Thus, the jammer can effectively assault via relay, i.e., the satellite, as the terminal is harder to target. The challenge is that the jammer must be close to the receiver, or it can enhance the detection potential.

In this paper, the authors develop a communication system model, the machine learning model, that learns the spectrum and makes transmission decisions, which automatically adapts to the spectrum dynamics. The generative adversarial learning neural network (GDNN) augments the synthesized training data based on the real data. It uses the generator and discriminator type neural networks with the minimax game theory to ensure that the transmission is successful. Paper's outline is given below: Section 2 provides the related works. Section 3 discusses the proposed model. Section 4 evaluates the GDNN model with existing methods. Section 5 concludes the entire work.

2 Related Work

In the following literature the effects of jamming were widely investigated for system performance and mitigation. However, it is not clearly addressed in literature how numerous antennas play a role in reducing jamming attacks under random data arrival.

In [11], the authors developed a module to reduce the effects of the jamming attack; the author created an intelligent adaptive sensing methodology which may also lessen sound effects during dynamic spectrum access (DSA) spectral sensing stages.

In [12], the authors developed a malicious protection system for node-based attacks to be mitigated in CRN-assisted agents. With the help of a certificate-aware authentication hash chaining mechanism, a network is prevented from attacks. The analysis of sensing reports from secondary users (SUs) and security association (SA) detects malicious SUs in the network. Malevolent nodes act as a node of support to alleviate network jamming.

In [13], the authors devised a primary user emulation detection technique for jamming attempts in cognitive radio. The suggested approach is based on the compressed signal coding in a dictionary which depends on the channel. In particular, the sparse code convergence patterns in accordance with the dictionary are utilized to identify between a jammer. The decision-making process is performed as a classification process based on learning.

In [14], the authors presented a rapid-forward cooperative transmission system in which a victim node hops in a nearby full duplex helper node which quickly transfers the symbol of the victim together with its symbol. The jammer and the aid worked together to give the opponent a portion of his power.

In [15], the authors presented a safe routing system which takes into account jamming of attacks which interrupt the transmission of cognitive radios. In accordance with an optimization problem, the suggested protocol provides the secure channel for source-destination pair. In addition, because CRN is more prone to threats, a second layer of defense will be presented for the Ensemble Jamming Detection. The peculiarity in the behavior of jamming attacks is determined.

3 Proposed Model

In this section, the system validates the channel statistics and background transmitter's behaviors to differentiate the benign and malignant cognitive radios present in a network [16,17]. Two types of neural network with the game theory improve the attacker detection process and data transmission [18,19].

The proposed model consists of the following components, as shown in Fig. 1.

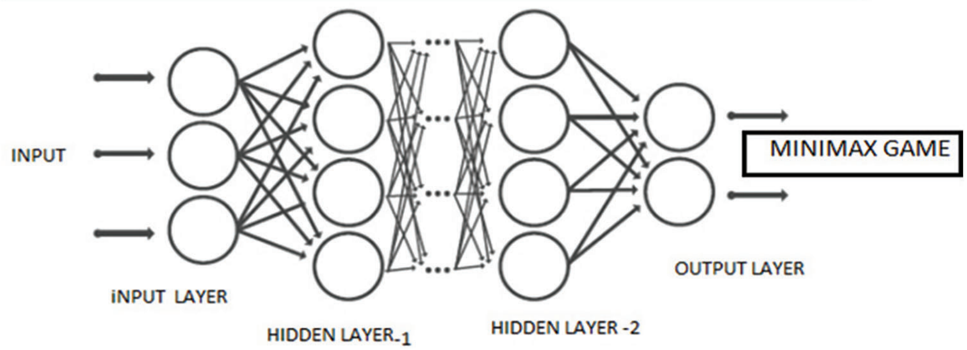


Figure 1: Neural network model with Minimax game as a proposed model

Algorithm: Detection of the malicious node

Step 1: Input Layer: Node's behavior and channel statistics trains the GDNN

Step 2: Neural Training: The deep neural network is trained with the loss function called cross-entropy function.

Step 3: Hidden Layer: Activation is performed using the sigmoid and hyperbolic tangent (Tanh) function.

Step 4: Output layer: The output layer uses the softmax activation.

Step 5: Minimax Game: The generator generates real data, and the discriminator player classifies the generated data.

In this work, a generative adversarial machine learning model has been used to detect the malicious node in the network effectively. It uses two types of neural networks, the generator and discriminator type neural networks, along with the minimax game theory, which facilitates the attacker detection and improves data transmission. Its working progress is represented in Fig. 2.

Here, GDNN is used with game theory for finding attacker node and eliminating as shown in the following Fig. 3.

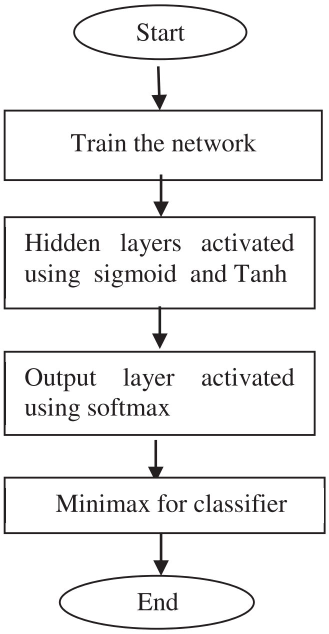

Start
Train the network
Hidden layers activated using sigmoid and Tanh
Output layer activated using softmax
Minimax for classifier
End

Figure 2: Proposed system using GAN with Minimax game theory

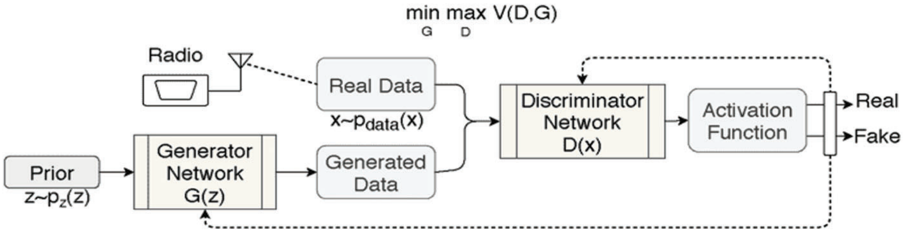


Figure 3: Working process of Minimax

where, D(x) is the discriminator's estimate the probability that read data distance x is real, E_x-is the expected value over the real data instances, G(z)–is the generator's output when given noise z, D(G(z))–is the discriminator's estimate of the probability that fake a instance is real, E_z-is the expected value over all random inputs to the generator (in effect, the expected value over all generated fake instances G(z)).

An execution of GAN with minimax game theory algorithm is shown in Fig. 4.

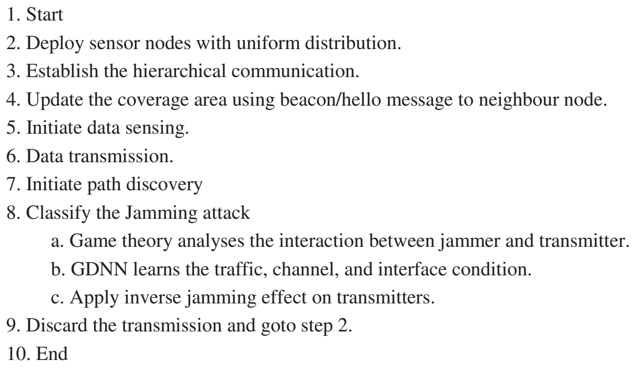
```
// Training loop
      for each training iteration do
            for k steps do
                  node m abnormal nodes $\{z_1, \dots z_m\}$ and transform with generator
                  node  m normal nodes  $\{x_1, \dots x_m\}$ from real data
                  update the discriminator by ascending the gradient
                  $\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} [logD(x^{(i)}] + \log(1 - D)\left(G\left(z^{(i)}\right)\right))]$
            end for
      node m noise abnormal nodes $\{z_1, \dots z_m\}$ and transform with generator
       update the generator by descending the gradient
                  $\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log(1 - D)\left(G\left(z^{(i)}\right)\right))$
       end for
```

Figure 4: Pseudocode of training loop for elimination of attacker node

This formula derives from the cross-entropy between the real and generated distributions. The generator can't affect the log(D(x)) term in the function, so, for the generator, minimizing the loss is equivalent to minimizing log(1−D(G(z))). The GAN loss function indicates that the generator tries to minimize the following function while the discriminator maximizes it.

Assume the transmission of data and jamming attack detection is executed in the wireless sensor network. The proposed approach working mechanism is as follows:

The cognitive radios [20,21] are deployed using uniform distribution in the network. Hierarchical communication is established between the sensor devices and the centralized base station. Each node updates its neighbor node [22,23] present in the coverage area using beacon/hello messages. Once the deployment and the neighbor communication are established data sensing and transmission are initiated. The path discovery process is involved to identify the available path between sensor and base station. The working progress of GDNN is shown in Fig. 5.



```
1. Start
2. Deploy sensor nodes with uniform distribution.
3. Establish the hierarchical communication.
4. Update the coverage area using beacon/hello message to neighbour node.
5. Initiate data sensing.
6. Data transmission.
7. Initiate path discovery
8. Classify the Jamming attack
    a. Game theory analyses the interaction between jammer and transmitter.
    b. GDNN learns the traffic, channel, and interface condition.
    c. Apply inverse jamming effect on transmitters.
9. Discard the transmission and goto step 2.
10. End
```

Figure 5: Pseudocode of GDNN–game theory

Jamming is performed by the attacker by generating more signal interference in the communication region to decrease successful transmission. The jammer applies an exploratory attack and studies the transmitter features of prior jamming. Under this attack, the jammer develops a classifier that represents a target classifier under attack. In a jammer, the transmissions or idle channels that tend to fail are not classified; instead, the jammer predicts the successful transmission, and jams it. The jamming of successful transmissions has two main objectives: (1) reducing the likelihood of misdetection and (2) reducing the likelihood of false alarms. Here, the transmit power is adjusted reference to the average power constraint. The selection of transmission power acts as a function of classification score at a time instant that tends to measure the probability of jamming attack as shown in Fig. 6. The malicious node detection is performed by invoking the GDNN game theory process.

4 Jammer Detection Using GDNN with Game Theory

Game theory in [24,25] general analyzes the conflicting communication between cognitive source transmitters and the jammers. During the transmission process GDNN learns the traffic condition, and interference in the channel that leads to success of data transmission between the radios. It further develops an inverse jamming strategy that increases the performance of communication [26,27]. The following input data for the neural network is collected at the time of data transmission after the data collection, and it includes the following:

- Packet forwarding count.
- Packet sent count.
- Drop counts of the data forwarder.
- Received signal strength (RSS) of each incoming data flow and its variations.
- Packet drop speed.
- Behavioral variations in radio and it is identified as:
- Average packet per flow count.
- Bytes per flow count and
- Rate of Flow Entries.

```
1. Allow jamming attacks to reduce the transmission.
2. Select the transmit power at any time slot.
3. Measure the likelihood of the jamming probability.
        a. Apply exploratory attack prior jamming.
        b. Build a classifier.
        c. Discard idle channels.
        d. Predict the nature of transmission.
        e. If the transmission is a success.
4. Jam the transmission
        a. else
5.  Do nothing
        a. end
6.  Reduce the misdetection probability.
7. Reduce the false alarm probability.
```

Figure 6: Pseudocode of jamming attack profile

From the collected data, the generator tends to generate the synthetic data with real-time samples in a short duration. From the estimated parameters, the Gaussian Kernel function and the linear polynomial kernel function values are calculated to differentiate the legitimate nodes from the attacker nodes in the network. The consumption is also derived with the Eigen function for each parameter. The Gaussian Kernel function is illustrated as follows,

$$f(x) = \frac{1}{n\sigma\sqrt{2\pi}} \sum_{i=0}^{n} e^{-\frac{1}{2}\left(\frac{x_i - x}{\sigma}\right)^2} \tag{1}$$

where n–number of sample, x_i, i$^{\text{th}}$ data sample,

The corresponding eigen function is defined as:

$$e^{-\varepsilon^2(x-z)^2} = \sum_{n=0}^{\infty} \lambda_n \varphi_n(x) \varphi_n(z) \tag{2}$$

$$\lambda_n = \frac{\alpha\varepsilon^{2n}}{\left[0.5\varepsilon^2\left[1 + \sqrt{1 + \left[\frac{2\varepsilon}{\alpha}\right]^2 + \varepsilon^2}\right]\right]^{n+0.5}}, n = 0, 1, 2$$

$$\varphi_n(x) = \frac{\sqrt[8]{1+\left(\frac{2\varepsilon}{\alpha}\right)^2}}{\sqrt{2^n}n!} e^{-\left(\sqrt{1+\left(\frac{2\varepsilon}{\alpha}\right)^2}-1\right)\frac{\alpha^2 x^2}{2}} \qquad \rho\left[\sqrt[4]{1+\left(\frac{2\varepsilon}{\alpha}\right)^2}\alpha x\right]$$

$\rho(x) = \frac{\alpha}{\sqrt{\Pi}} \; e^{-\alpha^2}x^2$ x-data smaple, z–mean, α-distribution factor (0,1), n-number of samples, ε-upper bound on the relative error.

The polynomial kernel between the two parameters is computed as follows,

$$K(a,b) = \left[1+\sum_j a_j b_j\right]^d \tag{3}$$

d-Indicates the degree.

4.1 GLANN

The cumulative sum of the weighted input is computed as a trained value for each set of collected data that represents the node behaviors. The mean value of the training value is computed from the entire set of trained values, and it is represented as the predictor. The behavioral pattern formation for each parameter is computed by taking the individual parameter as input. The neural training is conducted with the cross-entropy loss function with back-propagation using the following Eq. (4)

$$C(\theta) = -\sum_i \left(1-[y_T]_i\right)\log\left(1-\left[a^L(x_T)_i\right]\right) + [[y_T]_i \log\left(\left(\left[a^L(x)_T\right]_i\right)\right) \tag{4}$$

where, θ–neural network parameters, x_T-training data vector, y_T–label vector, $a^L(x)_T$-neural network output.

In hidden layer, sigmoid and hyperbolic tangent (Tanh) functions are used for the activation and it is defined as Eq. (5)

$$[\sigma(=)]_k = \frac{1}{1+e^{-z_k}}, Tanh = \frac{e^{-z_k}-e^{-z_k}}{e^{-z_k}+e^{-z_k}} \tag{5}$$

where (z)–sigmoid function, Tanh is the hyberbolic tangent function, z–input, k-entry count.

The output layer is activated using the softmax activation function with the gradient descent and it is denoted in the Eq. (6)

$$[\sigma(=)]_k = \frac{e^{z_k}}{\sum_j e^{z_j}} \tag{6}$$

4.2 Game Theory Formalization on GAN

Once the output value is activated the minimax game is invoked between the generator and discriminator. The game strategy is illustrated in the Eq. (7)

$$min_G max_D \mathbb{E}_{x\sim p_{data}}[\log(\mathrm{D}_x))] - \mathbb{E}_{\mathrm{z}\sim \mathrm{p_z}}[\log(1-\mathrm{D(G)}))] \tag{7}$$

where z-input error, p_{data}–data distribution, D–Discriminator and G–Generator function.

Upon completion of the minimax game, the activated output represents the behavioral difference between the normal and attacker node. This will classify the attacker node and eliminate it from the data communication.

5 Results and Discussion

This section evaluates the efficacy of jamming attack detection and model using GDNN-game theory under different network conditions. The running environment is setup with the following parameters as in Tab. 1. The proposed method is tested in terms of various network metrics that includes energy consumption, residual energy, packet delivery ratio, NRO, delay (ms), throughput, jitter, goodput, packets dropped, relative energy, and network lifetime. The proposed method is compared with existing methods in terms of simulation time and packet generation interval to test how well the systems respond to the jamming attacks at the transmitter. The proposed GDNN-game theory (short GDNN) is compared with other existing methods that include mitigating stealthy jamming attacks (MJSA) and stealthy data transmission with deep learning (SDTDL) [28].

Table 1: Simulation parameters

Parameter	Value
No of nodes	100 sensors
Base station	1
Queue type	Priority queue
Mac type	Sensor mac with IEEE 802.11
Topology area	500 × 500
Antenna type	Omni directional
Coverage area	80 m
Connection type	UDP
Packet size	512
Packet generation interval	0.1,0.2,0.3,0.4,0.5
Application type	Sensor (Temperature)
Simulation time	100–500 s
Data rate	Mbps
Initial energy	100 J
No of attacker	4 jamming attackers

5.1 Performance Evaluation

According to the simulation, game adversarial network (GAN) is used to learn about the transmission properties, as well as predict the possibility of attacks at an early stage; the system has higher success rates and improved network characteristics. A GDNN-game detection model is evaluated under different network conditions in this section to determine its effect and compare it to other methods.

5.1.1 Total Remaining Energy

It refers to the total energy available of all nodes in the network after completing the data transmission, and it is computed by subtracting the consumed energy from the initial energy as shown in Figs. 9 and 24.

5.1.2 Total Consumed Energy

It refers to the total consumed energy of all nodes in the network after computing the data transmission, and it is computed as shown in Figs. 7 and 22.

5.1.3 Throughput

Its outcome is the number of bits transmitted per unit of time. It is calculated from the size of the data packet, and the total number of received data packets with the transmission duration, as shown in Figs. 15 and 30.

5.1.4 Relative Energy

It refers to the average energy required to complete the unit data transmission, as shown in Figs. 20 and 35.

5.1.5 Packet Received

It indicates the total number of successful transmission completed in the network as shown in Figs. 11 and 26.

5.1.6 Number of Dropped Packets

Number of dropped packets: It is computed as the difference between the number of attempted packet transmissions and the number of successful packet transmission, as shown in Figs. 18 and 33.

5.1.7 Packets Distribution Ratio (PDR)

PDR is the ratio between the number of packets successfully received and number of packets attempted for transmission. It represents the success ratio of transmission over the wireless medium, as shown in Figs. 12 and 27.

5.1.8 Normalized Routing Overhead (NRO)

It represents the number of control messages required to transmit a single data packet, and it is calculated from the value of dividing the total number of received packets by the total number of control overhead messages, as shown in Figs. 13 and 28.

5.1.9 Lifetime

Time taken to drain the total energy of the nodes based on the transmission, and it is computed as shown in Figs. 21 and 36.

5.1.10 Jitter

It refers to the average consecutive packet transmission delay as shown in Figs. 17 and 32.

5.1.11 Goodput

It is the ratio of the total number of data bits transmitted for the entire simulation to the time taken to complete the overall transmission, as shown in Figs. 16 and 31.

5.1.12 Dropping Ratio

It is the ratio between the number of failed packet transmissions and number of attempted packet transmission, as shown in Figs. 19 and 34.

5.1.13 Delay

It refers to the average time taken to complete the end-to-end transmission of data packets from source to destination in the network, as shown in Figs. 14 and 29.

5.1.14 Average Remaining Energy and Average Consumed Energy

It refers to the average energy available of all nodes in the network after completing the data transmission and it is computed by subtracting the consumed energy from the initial energy as shown in Figs. 10 and 25. Average consumed energy refers to the average consumed energy of all nodes in the network after completing the data transmission, and it is computed as shown in Figs. 8 and 23.

5.1.15 Attacker Detection Ratio and Detection Delay

It represents the total number of attackers detected from the attacker presents in the network. Detection delay refers to the average time taken to detect the node from the time of attacker launch, as shown in Tab. 2. Tabs. 3 and 4.

The proposed model includes components such as sensor nodes, base station, neural network and input and output layers. The results of a GDNN based jamming detection are presented in which parameters such as attacker detection ratio and detection delay metrics are compared with other method such as MSJA and SDTDL and the best result is presented as the recommended method, as shown in Tabs. 2–4.

Table 2: Attacker detection ratio (Time based)

Simulation time	Attacker detection ratio			Detection delay		
	MSJA	SDTDL	GDNN	MSJA	SDTDL	GDNN
100	0.75	0.76	0.78	0.077958	0.068385	0.063257
125	0.78	0.8	0.82	0.076277	0.067006	0.062495
150	0.8	0.81	0.83	0.07512	0.066679	0.062244
175	0.815	0.82	0.84	0.07475	0.065882	0.061766
200	0.82	0.84	0.86	0.074512	0.065753	0.06172

Table 3: Attacker detection ratio (packet (Pkt) based)

Pkt generation interval	Attacker detection Ratio			Detection delay		
	MSJA	SDTDL	GDNN	MSJA	SDTDL	GDNN
0.1	0.79	0.8	0.82	0.074512	0.065753	0.06172
0.2	0.82	0.84	0.86	0.077123	0.069165	0.063159
0.3	0.84	0.85	0.87	0.080139	0.07216	0.065293
0.4	0.856	0.86	0.88	0.083445	0.074047	0.06838
0.5	0.86	0.88	0.9	0.084176	0.075807	0.067319

Table 4: Attacker detection ratio (Node based)

Number of nodes	Attacker detection ratio			Detection delay		
	MSJA	SDTDL	GDNN	MSJA	SDTDL	GDNN
50	0.77	0.78	0.8	0.09935	0.87671	0.082293
100	0.8	0.82	0.84	0.101249	0.090032	0.083283
150	0.82	0.83	0.85	0.103506	0.092559	0.085025
200	0.84	0.84	0.86	0.106481	0.094035	0.08725
250	0.86	0.88	0.9	0.108089	0.096127	0.08705

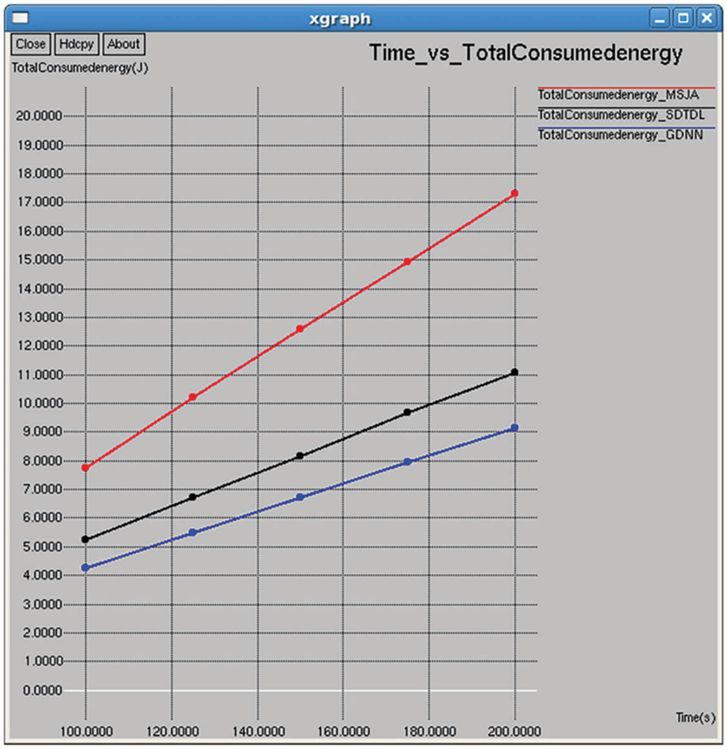


Figure 7: Total consumed energy

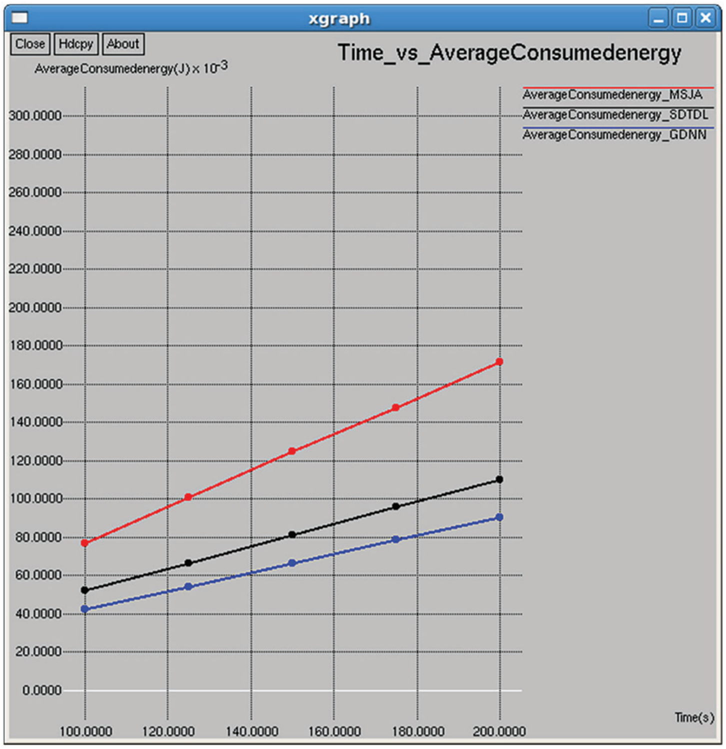


Figure 8: Average consumed energy

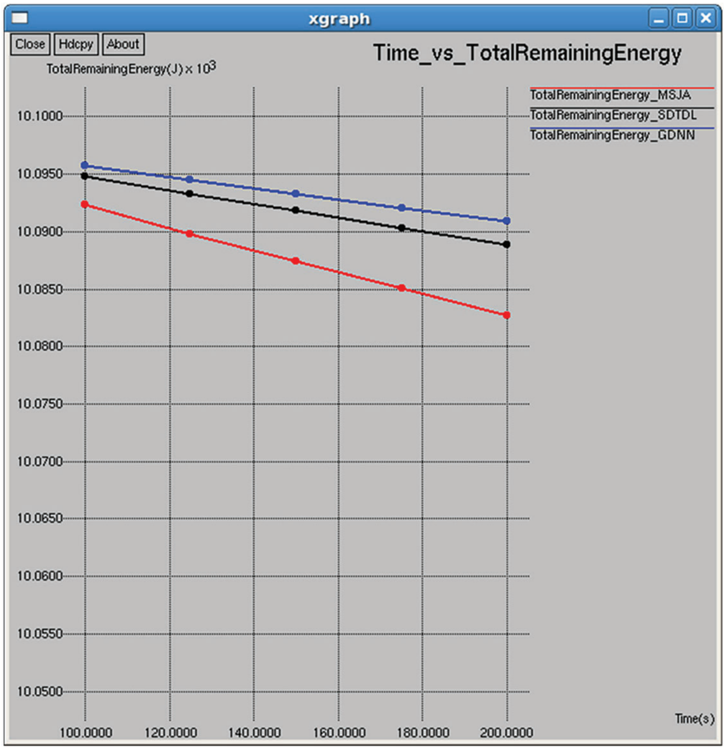


Figure 9: Total remaining energy

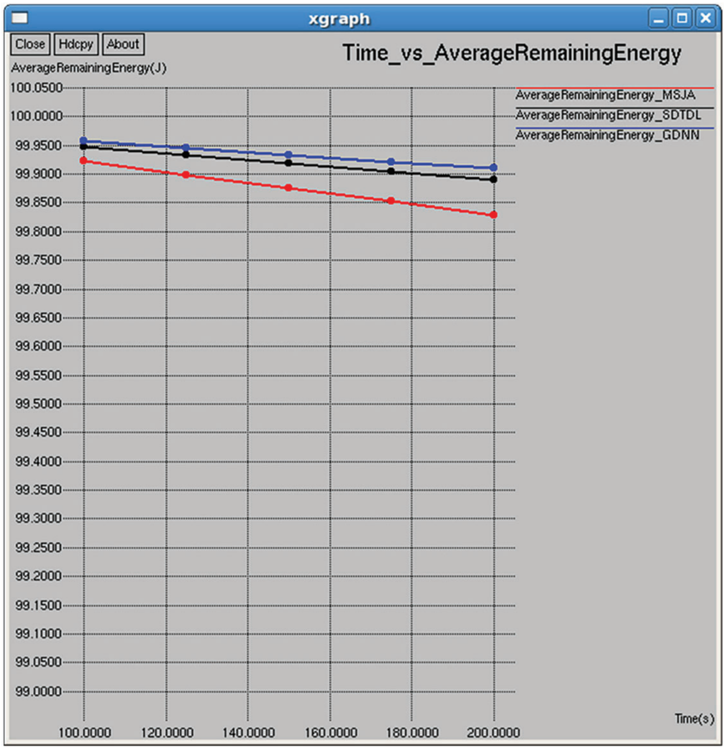


Figure 10: Average remaining energy

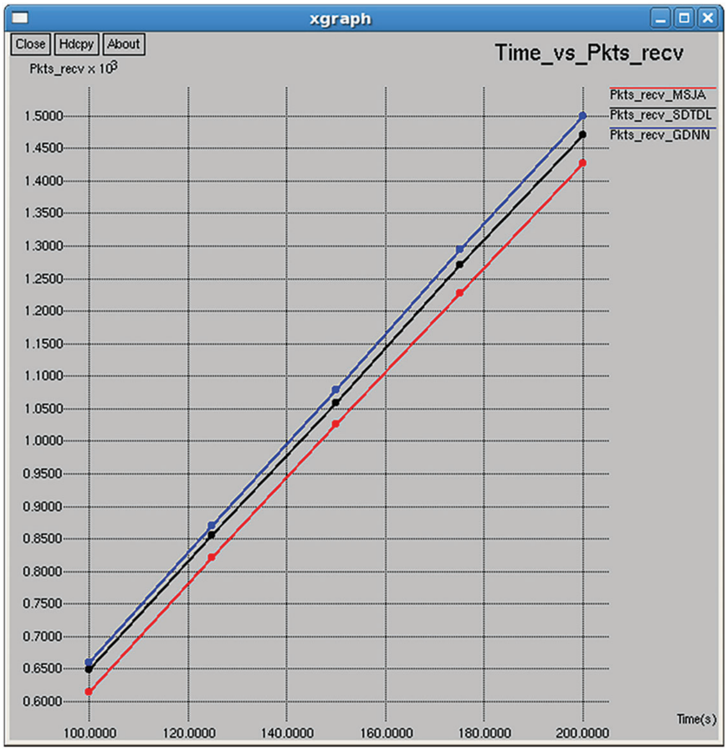


Figure 11: Total packets received

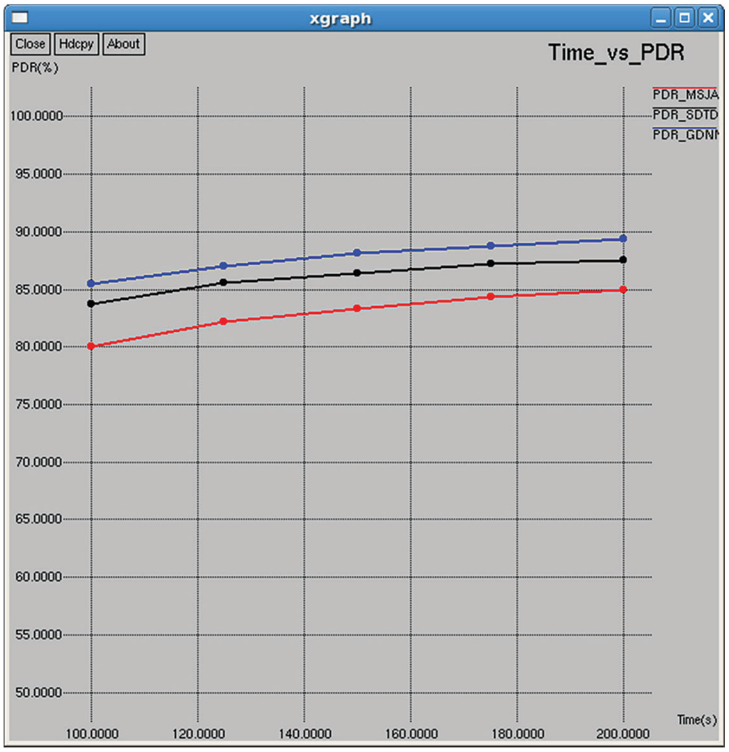


Figure 12: Packet delivery ratio

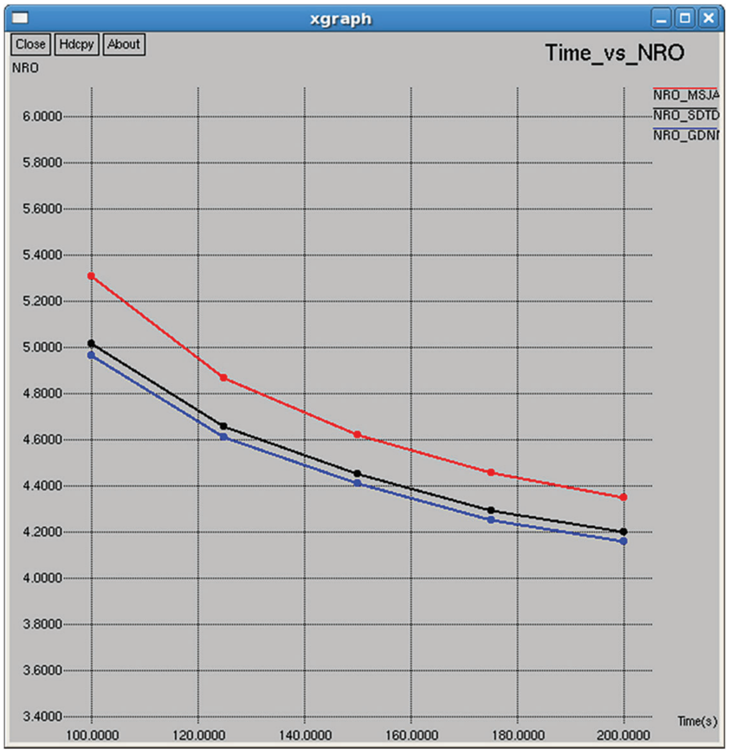


Figure 13: NRO

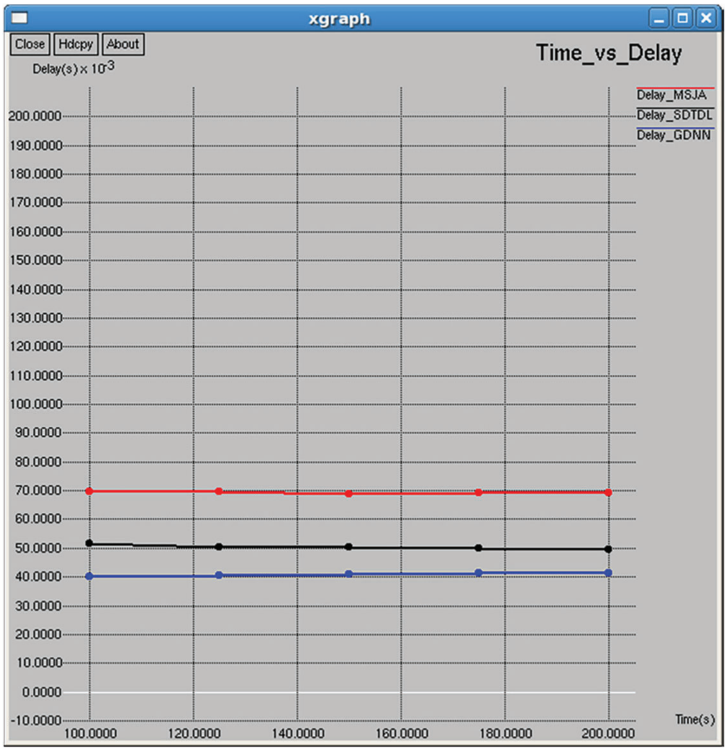


Figure 14: Delay (ms)

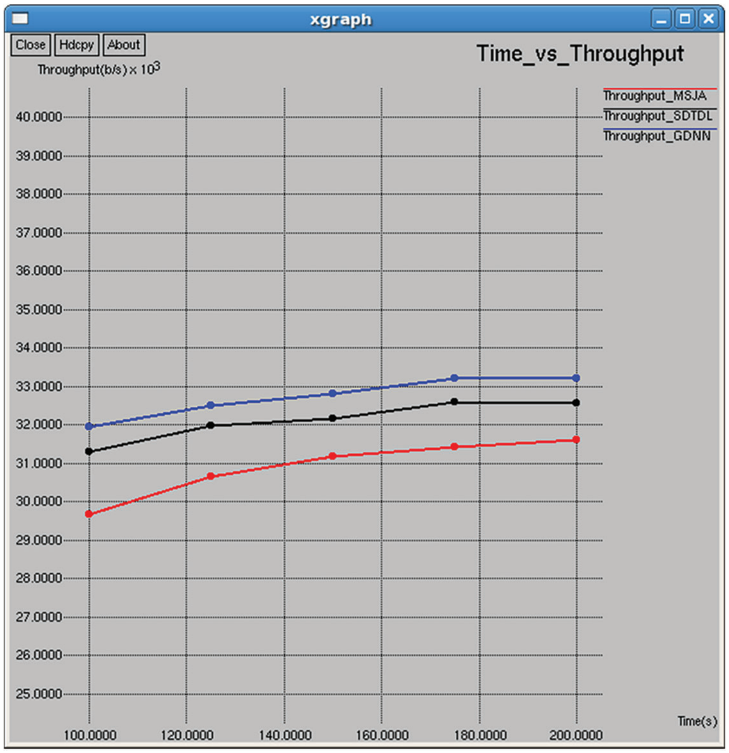


Figure 15: Network throughput

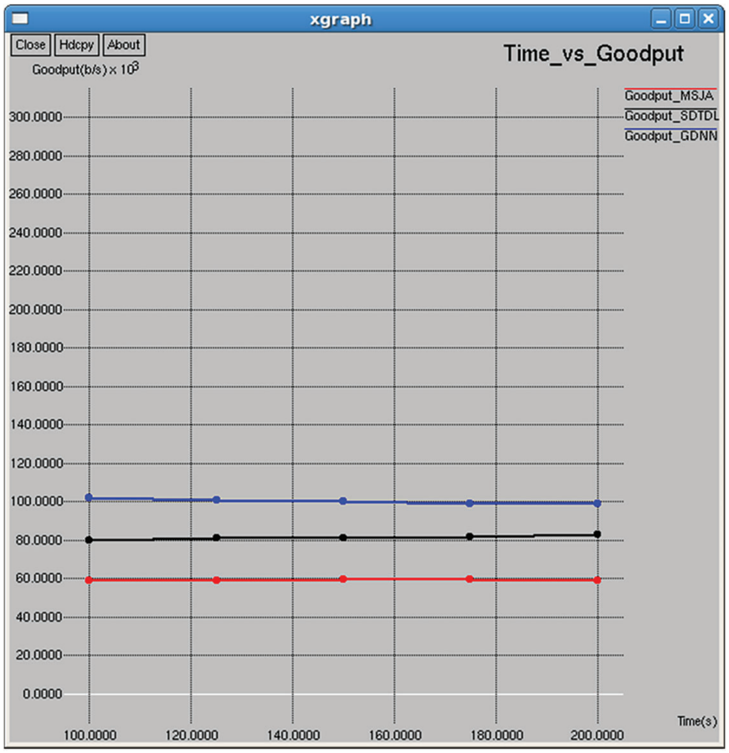


Figure 16: Goodput

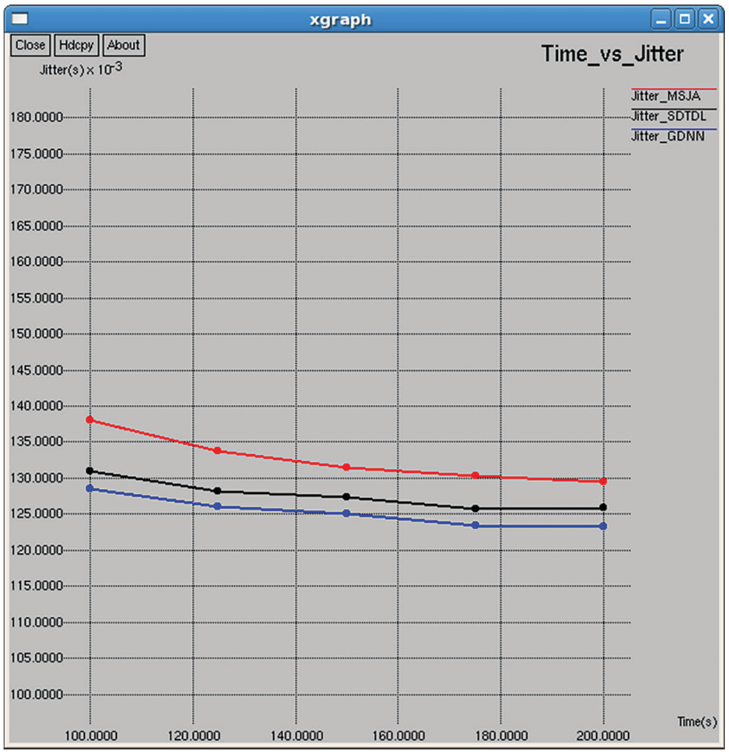


Figure 17: Jitter

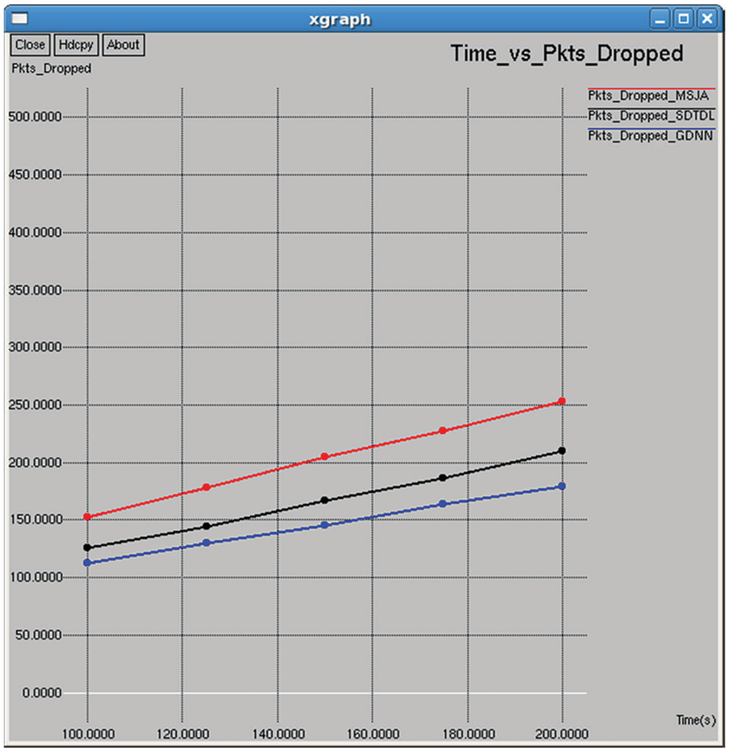


Figure 18: Packets dropped

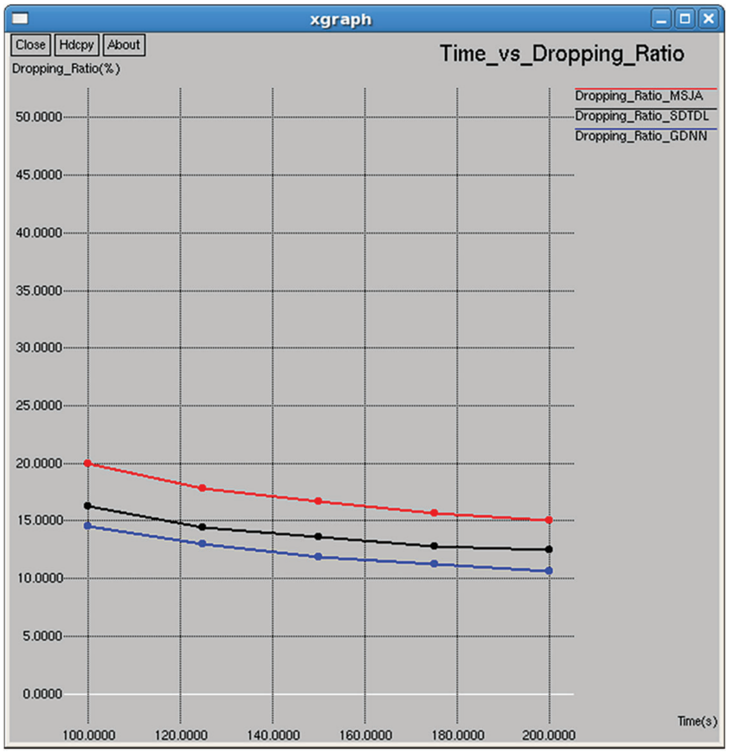


Figure 19: Dropping ratio

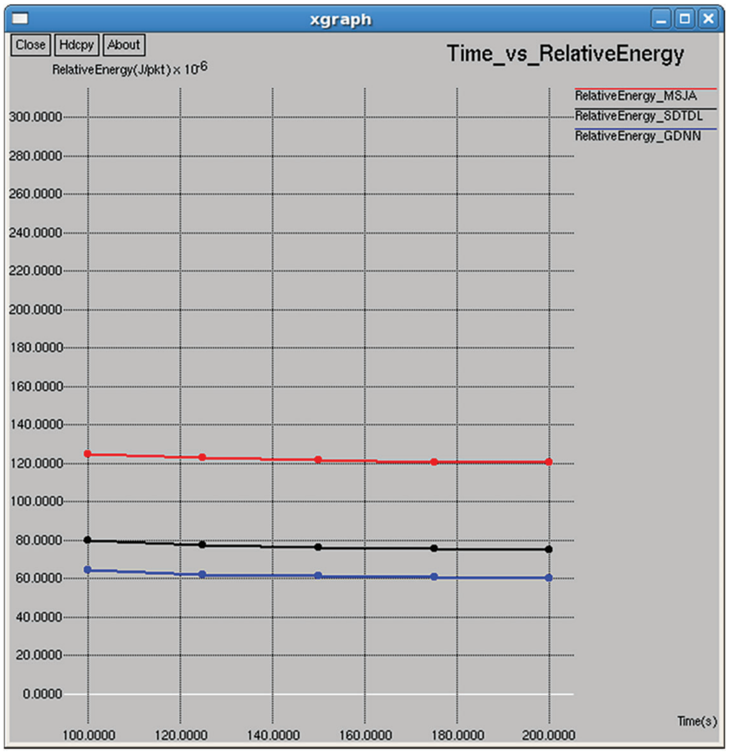


Figure 20: Relative energy

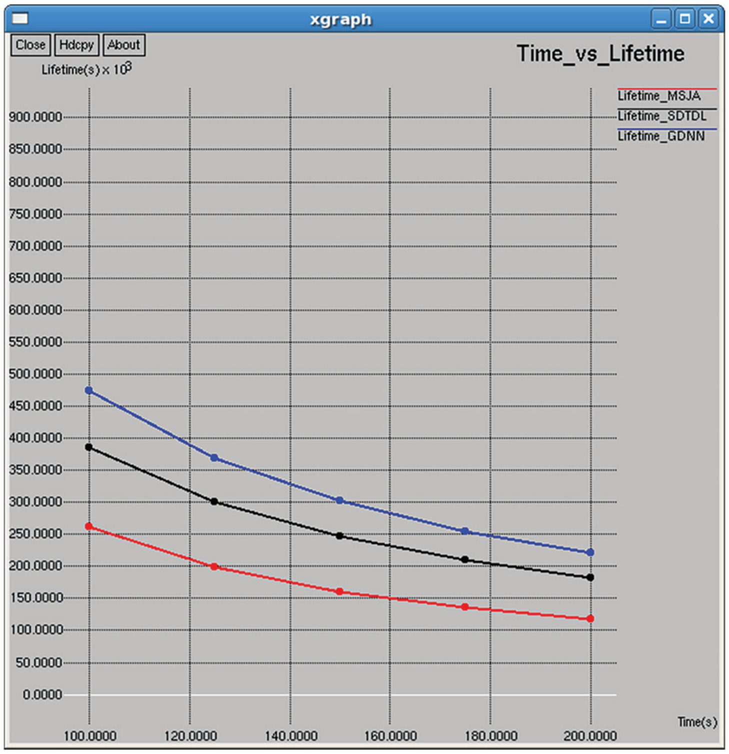


Figure 21: Life time

The packet delivery ratio has improved, and the routing overhead has been minimized. On comparison with MSJA and SDTDL it improves the lifetime of the network. Metrics jitter measures the delay between packets. It must be possible to transmit without any hindrance if the delay between transmission is negligible.

The disseminated learning network is used to find deviations in the behavior of nodes. This leads to a reduction in jitter and packet loss, as well as an increase in throughput.

The trained deep neural network validates the characteristics of each node. The neural network output is the nodes status. The node is removed from data transmission if it is determined to be malicious.

In order to initiate the data transmission, the route from the source node to the destination must be determined. The routes have to be identified and sensed. The data transmission is initiated after the route has been identified. An attacker node is detected and removed from the network during the transmission of data. For evaluating the performance of suggested methodology, certain metrics such as jitter, goodput, throughput, packet dropped, packet delivery ratio are proposed.

Good performance contributes to the amount of information delivered and the packet dropping ratio must be low. This result of metrics shows that the GDNN is capable of successful transmission of the metrics.

Here we will present graphical representations of the comparison of various metrics with time, as well as compare various metrics with packet generation intervals.

5.2 Comparison of Various Metrics w.r.t Time

In this section, we present the simulation results on various performance metrics between the GDNN and existing methods. The results show reduced total, and average consumed energy, reduced total and average residual energy, increased packets received with packet delivery ratio reduced packets dropped with reduced packet dropping ratio, reduced delay and jitter, and increased goodput and network lifetime than existing methods. Further, it is seen that the utilization of the attack detection model using GDNN enables optimal transmission of data packets successfully between the source and base station (BS). However, the case is not true in existing methods, the entire computational efficiency is spent on detecting the attack model rather than successfully sending the packets to BS from the source transmitter.

5.3 Comparison of Various Metrics w.r.t Packet Generation Interval

This section provides the results of various performance metrics evaluated between the proposed and existing methods in terms of packet generation interval. The simulation results are similar to those that appeared w.r.t to the training interval. However, it is seen that the total and average consumed energy reduced with increasing packet generation interval (PGI) are against the simulation time, where consumption increases with time. The same is the case of the total packets received, network throughput and packets dropped. On other hands, it is seen that with increasing PGI, the following parameters increases total and average residual time, packet delivery rate, NRO delay, jitter, etc., against simulation time, where it acts in reverse manner.

From the simulation, it is inferred that the utilization of GAN is used to learn the transmission properties and predict the possibility of attacks at the earliest manner and ensures that the system obtains increased network characteristics with a higher successful transmission rate than other methods.

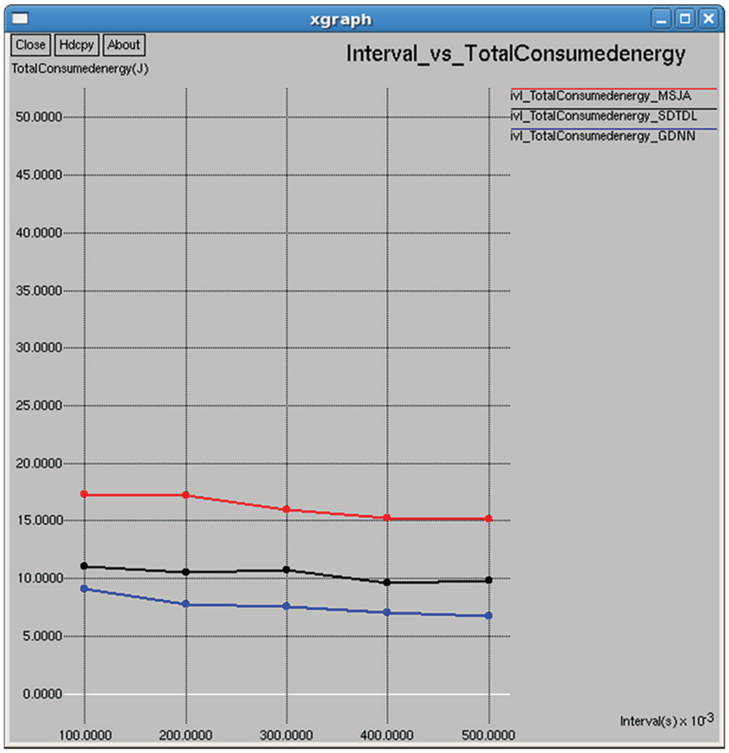


Figure 22: Total consumed energy

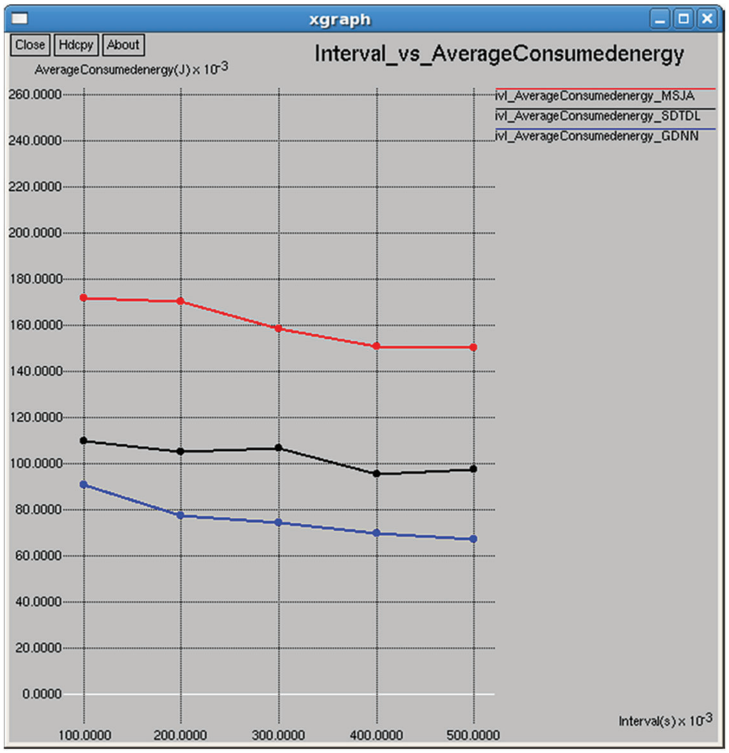


Figure 23: Average consumed energy

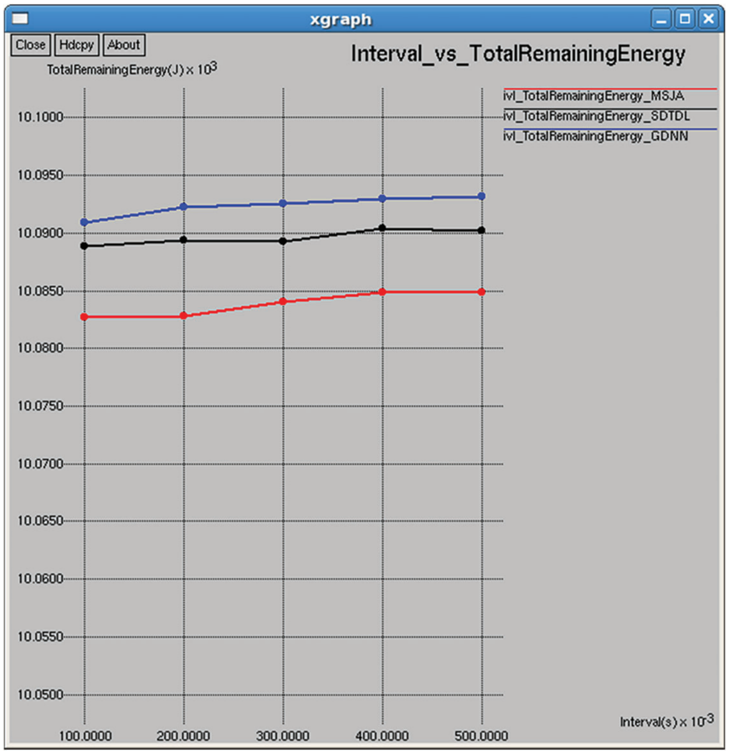


Figure 24: Total remaining energy

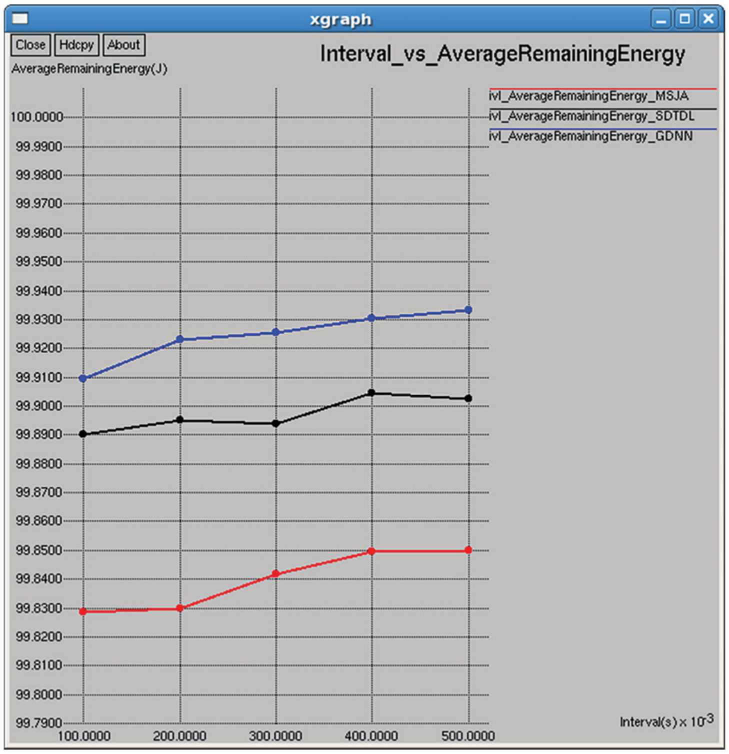


Figure 25: Average remaining energy

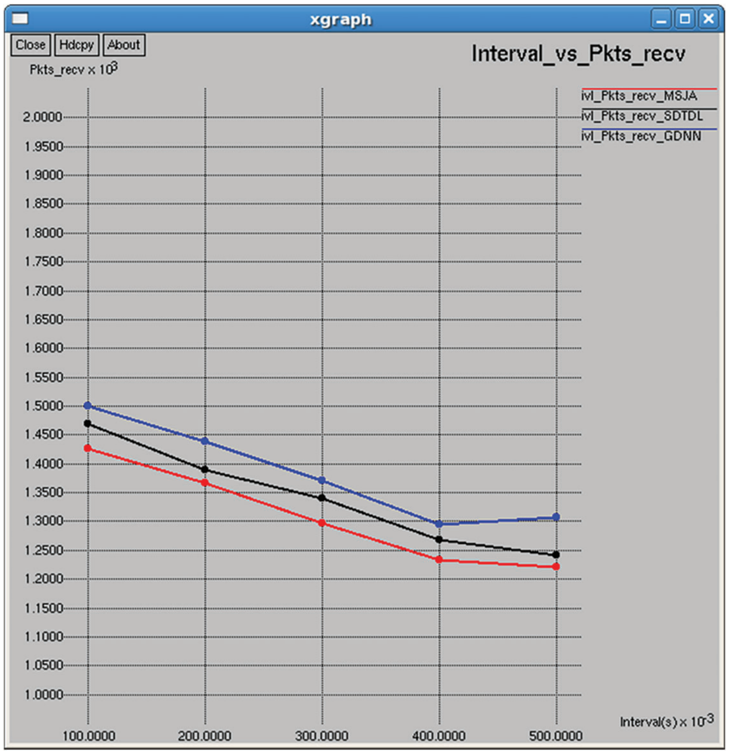


Figure 26: Total packets received

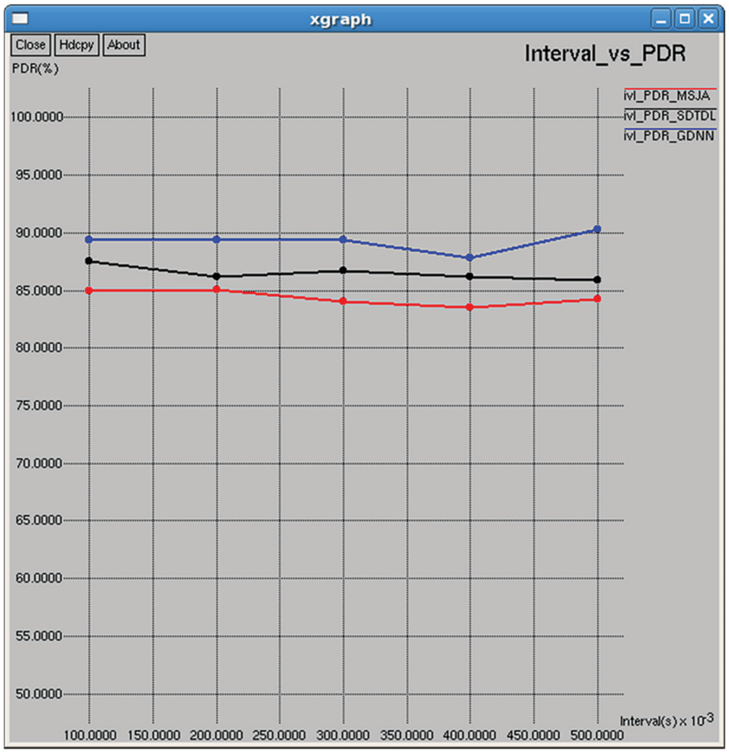


Figure 27: Packet delivery ratio

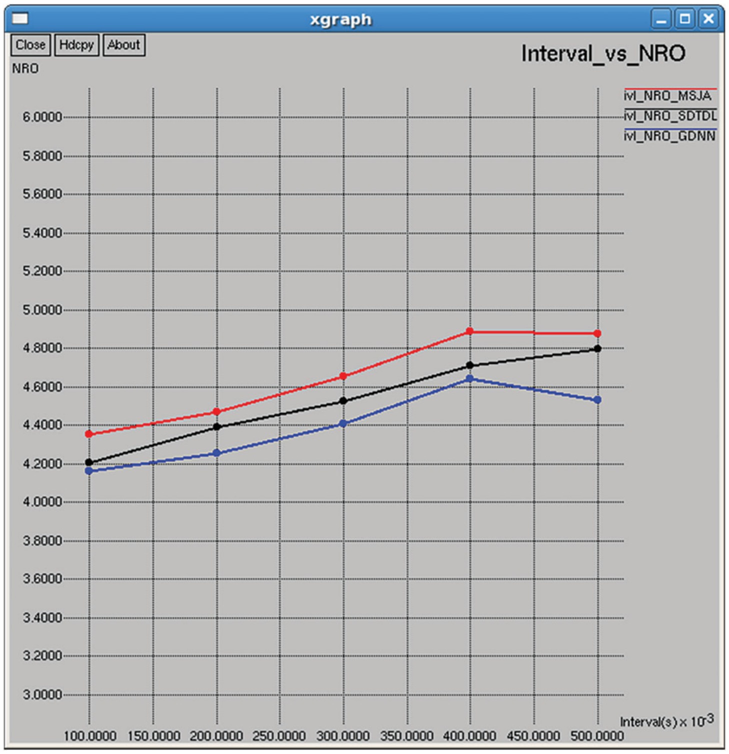


Figure 28: NRO

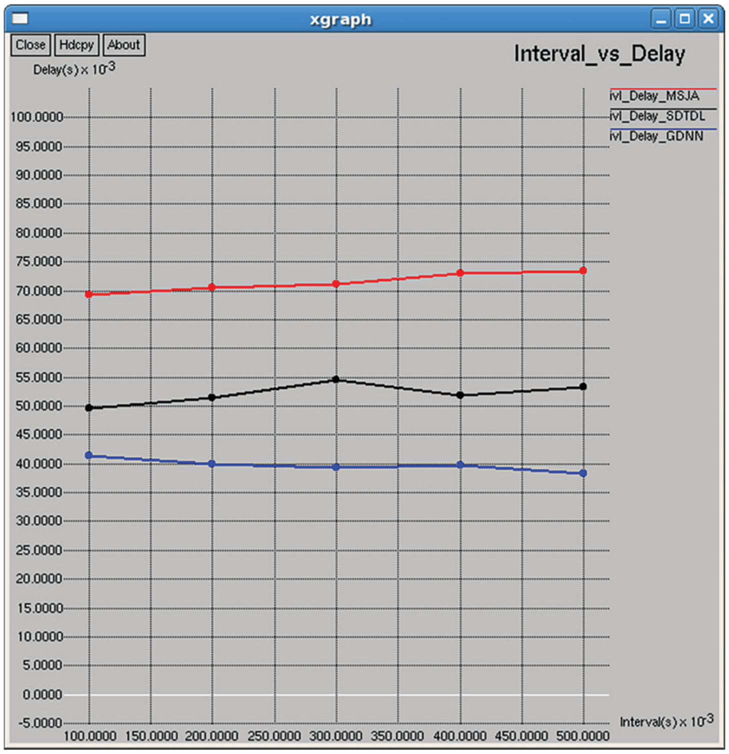


Figure 29: Delay (ms)

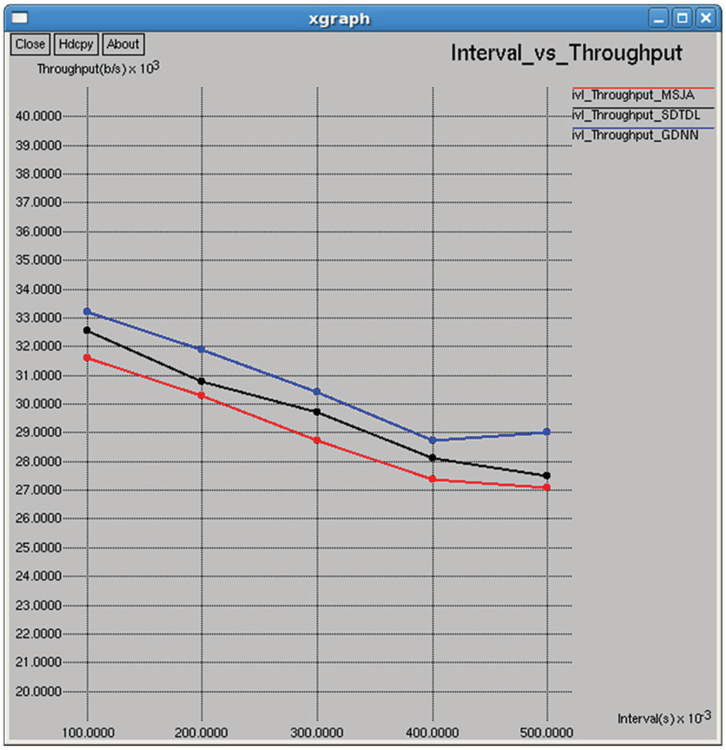


Figure 30: Network throughput

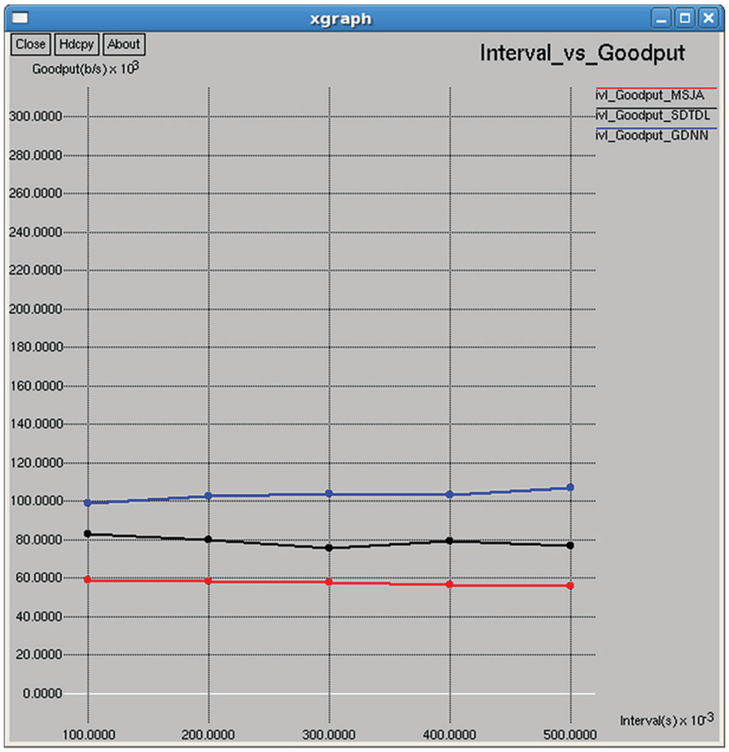


Figure 31: Goodput

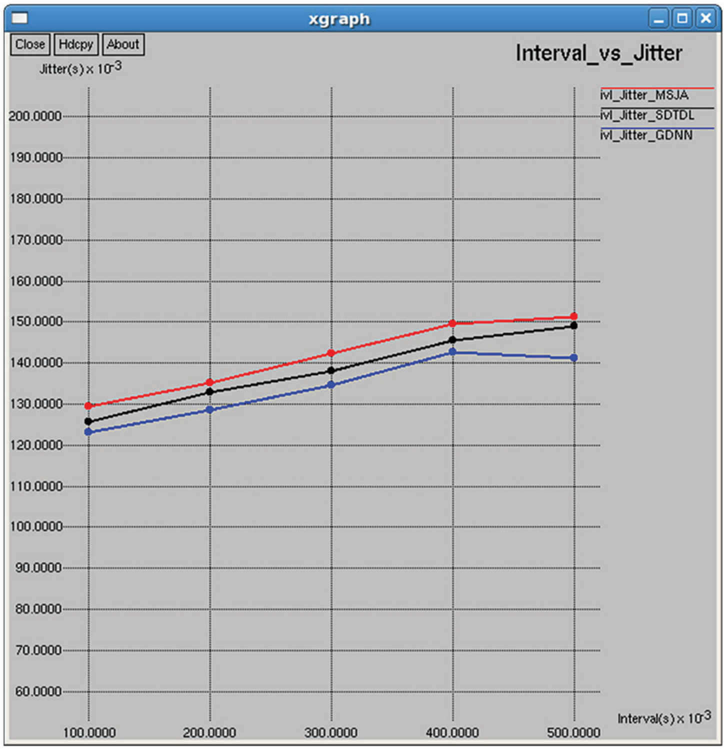


Figure 32: Jitter

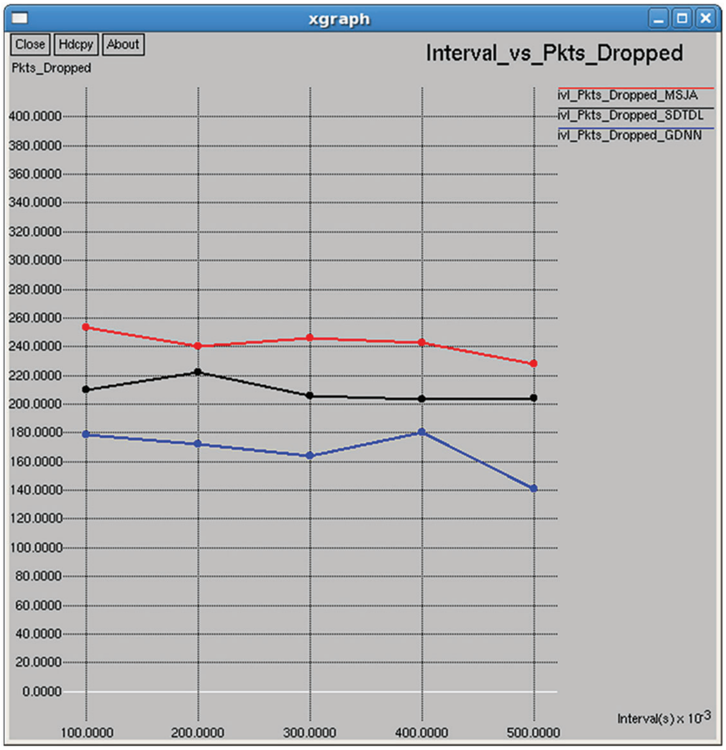


Figure 33: Packets dropped

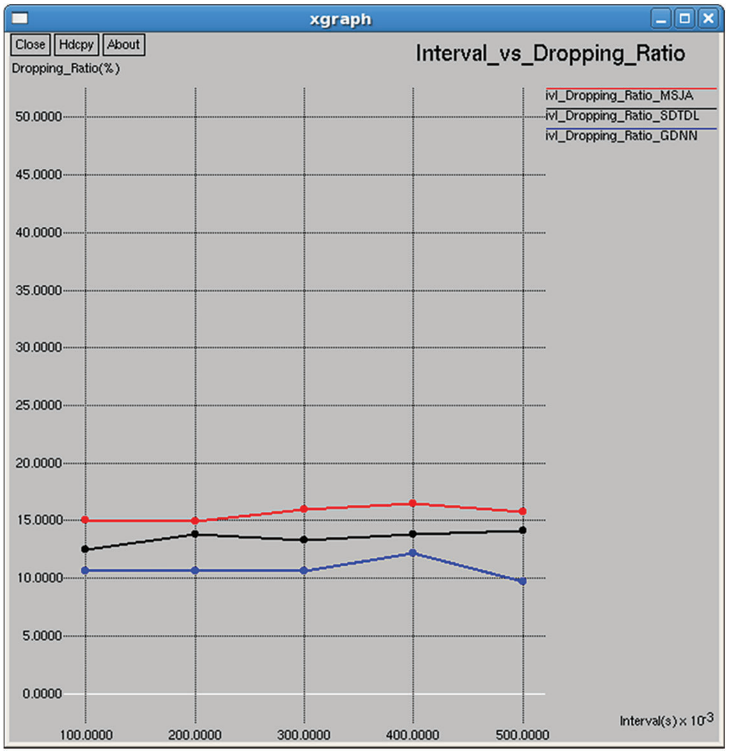


Figure 34: Dropping ratio

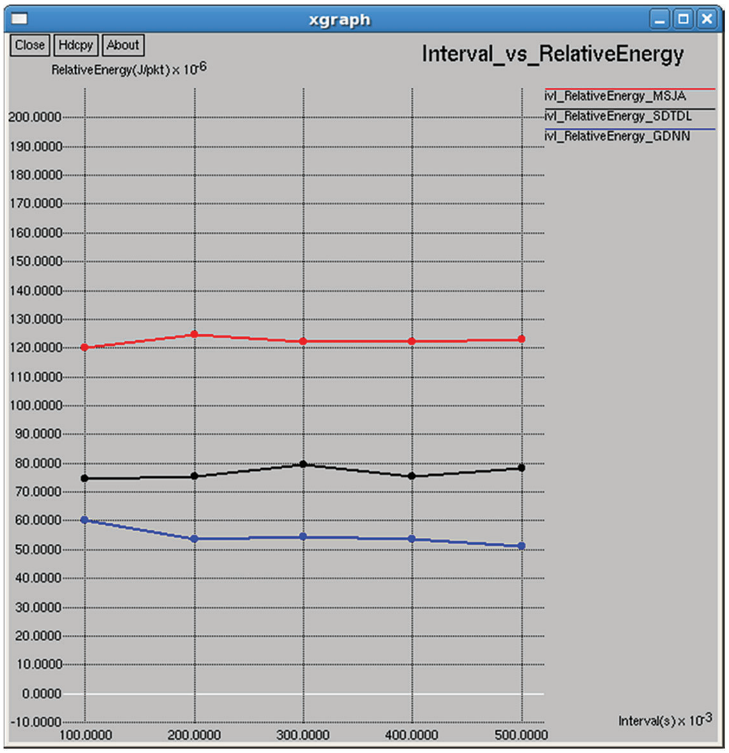


Figure 35: Relative energy

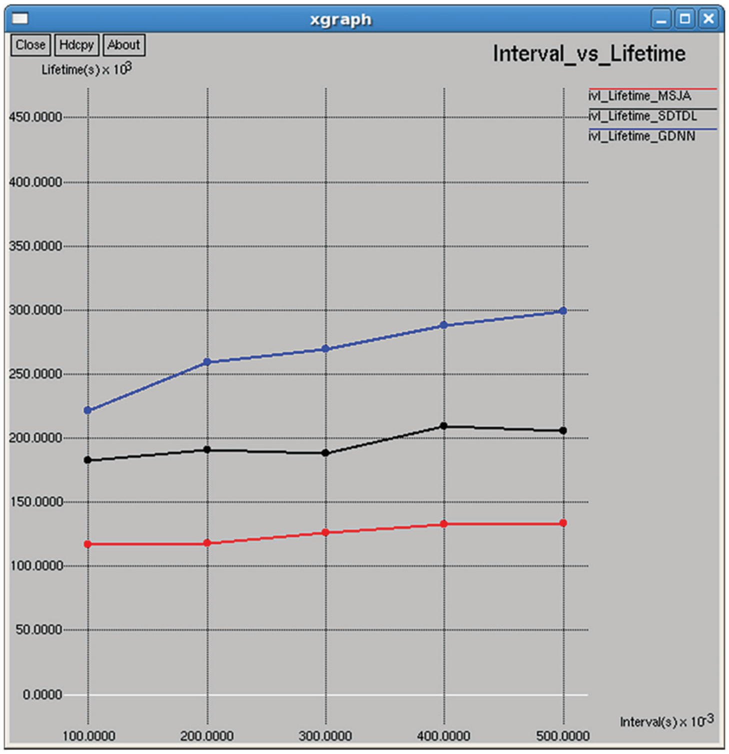


Figure 36: Lifetime

6 Conclusion

In this paper, we model a GDNN to improve the success of the transmission rate between source and BS by mitigating jamming attacks. GDNN with generator and discriminator type neural networks with the minimax game theory automatically adapts to the spectrum dynamics. The training of GDNN with such a defense mechanism misleads the jammers to attack the transmission of data. Such misleading via game theory does not allow the jammers to select the time slot since it makes inaccurate predictions on classification sources. Such poor decisions by jammers prevent major transmission losses and make the model efficient in successfully transmitting data packets between the source and destination nodes.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] X. R. Zhang, W. Z. Zhang, W. Sun, H. L. Wu, A. G. Song *et al.,* "A real-time cutting model based on finite element order reduction," *Computer Systems Science and Engineering*, vol. 43, no. 1, pp. 1–55, 2022.

[2] Y. E. Sagduyu, R. A. Berry and A. Ephremides, "Jamming games for power controlled medium access with dynamic traffic," in *Proc. ISIT*, Austin,Texas, USA, pp. 1818–1822, 2010.

[3] W. Sun, G. C. Zhang, X. R. Zhang, X. Zhang and N. N. Ge, "Fine-grained vehicle type classification using lightweight convolutional neural network with feature optimization and joint learning," *Multimedia Tools and Applications*, vol. 80, no. 20, pp. 30803–30816, 2021.

[4] Y. E. Sagduyu, R. A. Berry and A. Ephremidesi, "Wireless jamming attacks under dynamic traffic uncertainty," in *Proc. 8th Int. Symp. on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, Avignon, France, pp. 303–312, 2010.

[5] B. Yin, S. W. Zhou, S. W. Zhang, K. Gu and F. Yu, "On efficient processing of continuous reverse skyline queries in wireless sensor networks," *KSII Transactions on Internet and Information Systems*, vol. 11, no. 4, pp. 1931–1953, 2017.

[6] M. Karlsson, E. Bjornson and E. G. Larsson, "Jamming a TDD point-to-point link using reciprocity-based MIMO," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 12, pp. 2957–2970, 2017.

[7] J. M. Zhang, K. Yang, L. Y. Xiang, Y. S. Luo, B. Xiong *et al.,* "A self-adaptive regression-based multivariate data compression scheme with error bound in wireless sensor networks," *International Journal of Distributed Sensor Network*, vol. 9, no. 3, pp. 913497, 2013.

[8] M. Medard, "Capacity of correlated jamming channels," in *Proc. Annual Allerton Conf. on Communication Control and Computing, University of Illinois*, vol. 35, pp. 1043–1052, 1997.

[9] J. Wang, X. C. Ju, Y. Gao, A. K. Sangaiah and G. J. Kim, "A PSO based energy efficient coverage control algorithm for wireless networks," *Computers, Materials & Continua*, vol. 56, no. 3, pp. 433–446, 2018.

[10] S. Shafiee and S. Ulukus, "Capacity of multiple access channels with correlated jamming," in *Proc. IEEE MILCOM*, Atlantic, NJ, USA, pp. 218–224, 2005.

[11] M. F. Amjad, H. Afzal, H. Abbas and A. B. Subhani, "AdS: An adaptive spectrum sensing technique for survivability under jamming attack in cognitive radio networks," *Computer Communications*, vol. 172, no. 4, pp. 25–34, 2021.

[12] N. Saini, N. Pandey and A. P. Singh, "Developing malevolent node-based protection system against jamming attack in agent assisted CRN," *International Journal of Information and Computer Society*, vol. 13, no. 1, pp. 73–96, 2020.

[13] H. M. Furqan, M. A. Aygul, M. Nazzal and H. Arslan, "Primary user emulation and jamming attack detection in cognitive radio via sparse coding," *EURASIP Journal on Wireless Communications and Networking*, vol. 1, no. 1, pp. 1–19, 2020.

[14] V. Chaudhary and H. Jagadeesh, “Fast-forward mitigation schemes for cognitive adversary,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 4, pp. 1304–1319, 2021.

[15] H. B. Salmeh, S. Otoum, M. Aloqaily, R. Derbas, I. A. I. Ridhawi *et al.,* “Intelligent jamming-aware routing in multi-hop IoT-based opportunistic cognitive radio networks,” *Ad Hoc Networks*, vol. 98, no. 3, pp. 102035, 2020.

[16] J. Wang, X. J. Gu, W. Liu, A. K. Sangaiah and H. J. Kim, “An empower Hamilton loop based data collection algorithm with mobile agent for WSNs,” *Human-Centric Computing and Information Sciences*, vol. 9, no. 1, pp. 1–14, 2019.

[17] J. Wang, Y. Gao, C. Zhou, S. Sherratt and L. Wang, “Optimal coverage multi-path scheduling scheme with multiple mobile sinks for WSNs,” *Computers, Materials & Continua*, vol. 62, no. 2, pp. 695–711, 2020.

[18] J. Wang, Y. Gao, W. Liu, W. Wu and S. J. Lim, “An asynchronous clustering and mobile data gathering schema based on timer mechanism in wireless sensor network,” *Computers, Materials & Continua*, vol. 58, no. 3, pp. 711–725, 2019.

[19] J. Wang, Y. Gao, X. Yin, F. Li and S. J. Kim, “An enhanced PEGASIS algorithm with mobile skin support for wireless sensor networks,” *Wireless Communications and Mobile Computing*, vol. 2018, no. 8, pp. 1–9, 2018.

[20] Q. Tang, K. Yang, P. Li, J. M. Zhang, Y. S. Luo *et al.,* “An energy efficient MCDS construction algorithm for wireless sensor networks,” *EURASIP Journal on Wireless Communication and Networking*, vol. 2012, no. 1, pp. 102, 2012.

[21] Z. Liao, J. Wang, S. Zhang, J. Cao and G. Min, “Minimizing movement for target coverage and network connectivity in mobile sensor networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 7, pp. 1971–1983, 2014.

[22] J. Heo, J. J. Kim, J. Peak and S. Bahk, “Mitigating stealthy jamming attacks in low-power and lossy wireless networks,” *Journal of Communications and Networks*, vol. 20, no. 2, pp. 219–230, 2018.

[23] Y. Xi, L. Kong, Z. Liu, Y. Che, Y. Li *et al.,* “Machine learning and deep learning methods for cybersecurity,” *IEEE Access*, vol. 6, pp. 35365–35381, 2018.

[24] T. Erpek, Y. E. Sagduyu and Y. Shi, “Deep learning for launching and mitigating wireless jamming attacks,” *IEEE Transaction on Cognitive Communications and Networking*, vol. 5, no. 1, pp. 2–14, 2018.

[25] X. Wei and Q. Sun, “A jamming detection method for multi-hop wireless networks based on association graph,” *Int. J. High Performance Computing and Networking*, vol. 14, no. 3, pp. 284–293, 2019.

[26] J. Xu, H. Lou, W. Zhang and G. Sang, “An intelligent anti-jamming scheme for cognitive radio based on deep reinforcement learning,” *IEEE Access*, vol. 8, pp. 202563–202572, 2020.

[27] L. Zhao, H. Xu, J. Zhang and H. Yang, “Resilient control for wireless cyber-physical systems subject to jamming attacks: A cross-layer dynamic game approach,” *IEEE Transactions on Cybernetics*, vol. 52, no. 4, pp. 2599–2608, 2020.

[28] E. Jayabalan and R. Pugazendi, “Deep learning model-based of jamming attacks in low-power and lossy wireless networks,” *Soft Computing, Springer*, pp. 1–22, 2021. [Online]. Available: https://link.springer.com/article/10.1007/s00500-021-06111-7.