Details of a Digital Twin for a LoRa Based Forest Fire Management System

Buğra Aydın[1], **and Sema F. Oktuğ**[1]

[1] Department of Computer Engineering, Istanbul Technical University, Istanbul, 34469, Turkey

Abstract: Early detection of forest fires is vital for ecosystems. For this purpose, sensor networks collect data such as temperature and humidity and monitor changes in forests. Long-range and low-energy communication technologies such as LoRa are especially widely used in these networks. However, the management of these networks can be complicated since each forest has different requirements. Digital twin technology allows the simulation of different scenarios and optimization systems by creating virtual copies of physical systems to solve this problem. However, the relational structure of computer networks can be challenging for some artificial intelligence models used in digital twins. Graph neural networks help digital twins to understand and optimize the complicated structure of networks. In addition, it is not feasible for Internet of Things networks to meet digital twins' two-way and continuous communication demand. Therefore, in this study, a forecaster model is designed to facilitate the integration of digital twins into these networks. The forecaster provides the data the digital twin needs by predicting the network's future states from its past states. The first results of the study are promising, especially for small-scale networks. However, as the scale of the network grows, the errors made by the system also increase.

Keywords: Digital twin, IoT, graph neural networks, forest fire detection.

LoRa Tabanlı Bir Orman Yangını Yönetim Sistemi Dijital İkizinin Ayrıntıları

Özet: Orman yangınlarının erken tespiti, ekosistemler için hayati önem taşır. Bu amaçla sensör ağları, sıcaklık ve nem gibi verileri toplayarak ormanlardaki değişiklikleri izler. Özellikle LoRa gibi uzun menzilli ve düşük enerjili iletişim teknolojileri, bu ağlarda yaygın olarak kullanılır. Ancak bu ağların yönetimi, her bir ormanın farklı gereksinimleri olduğundan karmaşık olabilir. Dijital ikiz teknolojisi, bu sorunu çözmek için fiziksel sistemlerin sanal kopyalarını oluşturarak, farklı senaryoları simüle etmeye ve sistemleri optimize etmeye olanak tanır. Lakin bilgisayar ağlarının ilişkisel yapısı dijital ikizde kullanılan bazı yapay zeka modelleri için zorlayıcı olabilir. Grafik sinir ağları ise dijital ikizlerin, ağların karmaşık yapısını anlamasına ve optimize etmesine yardımcı olur. Ayrıca, nesnelerin interneti ağlarının, dijital ikizlerin iki yönlü ve sürekli iletişim talebini karşılaması uygulanabilir değildir. Bu nedenle, bu çalışmada dijital ikizlerin bu ağlara entegrasyonunu kolaylaştıracak bir tahminci modeli tasarlanmıştır. Tahminci ağın geçmiş durumlarından gelecek durumlarını tahmin ederek dijital ikizin ihtiyacı olan veriyi sağlar. Çalışmanın ilk sonuçları özellikle küçük ölçekli ağlar için umut vericidir. Ancak ağın ölçeği büyüdükçe sistemin yaptığı hatalar da artmaktadır.

Anahtar Kelimeler: Dijital ikiz, nesnelerin interneti, grafik sinir ağları, orman yangını tespiti.

RESEARCH PAPER

Corresponding Author: Sema F. Oktuğ, oktug@itu.edu.tr

Reference: B. Aydın, S. F. Oktuğ, (2025), "Details of a Digital Twin for a LoRa Based Forest Fire Management System," *ITU-Journ. Wireless Comm. Cyber.*, 2, (1) 27–36.

Submission Date: Mar, 2, 2025
Acceptance Date: Mar, 19, 2025
Online Publishing: Mar, 28, 2025

1 INTRODUCTION

The ecology of the world is crucially threatened by forest fires. Rising average temperatures have also increased the frequency of forest fires, including in Turkey. As indicated by the graph in Figure 1, there are more than 2 thousand forest fires every year, which destroy more than 10 thousand hectares of forest area [1].

Early and rapid-fire detection can significantly reduce the devastation of forest fires. Several different techniques are used for these detections such as surveillance of forests with satellites, flying over forests with Unmanned Aerial Vehicles (UAVs), and regularly monitoring forest values with wireless sensor networks. All of these systems have advantages and disadvantages. However, the most cost-effective and fastest fire detection method is Wireless Sensor Network (WSN) solutions. [2].

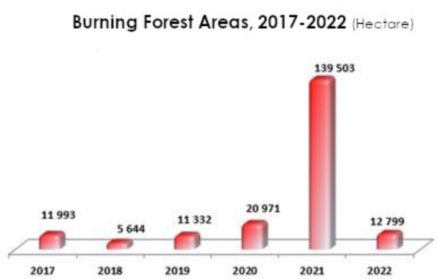


Fig. 1 Area of burned on every year.

Internet of Things (IoT) sensor networks are frequently used in these systems to detect forest fires early. However, forests differ from each other in many aspects, such as size, elevation difference, climatic conditions, tree density, and diversity. Therefore, the demands of forests and the structures of these networks vary widely. This requires specific decisions to be made in the management of each network. Also, due to the size of forests, managing these large networks can be difficult.

Once WSN networks are deployed in forests, their management becomes another problem to be solved. In these sophisticated networks, it may be desirable to minimize packet loss or optimize energy consumption. Digital Twin (DT) technology can help network administrators in this area. DTs are widely used in computer networks for optimization and running test cases. Nonetheless, DTs are challenging to use in IoT networks due to their constant communication requirements. Hence, this paper proposes a forecaster mechanism to facilitate this integration. The proposed model generates the data for DT by predicting the network traffic (packets) in advance.

This paper introduces a digital twin application for forest fire detection systems employing IoT networks. Integrating the digital twin into such networks is key to perceiving their complexity. It also enables effective and accurate testing of different strategies for network optimization. However, the main challenge in achieving this integration is the need for continuous bidirectional data transfer of digital twins. Since IoT networks have limitations in terms of energy and performance, meeting these requirements is challenging for them. In addition, some use cases can present interesting contradictions. For example, a digital twin modeling network packet delivery rate needs real-time lost packet data from the network. However, the digital twin is not aware of a packet that has not reached the network's server. In networks such as the one studied in this work, where forest fires are specifically investigated, LPWAN communication technologies are widely preferred. The additional limitations of these technologies, such as two-way communication, can make the integration of DTs even more intractable.

Instead of providing real-time network data to the DT, designing a forecaster that predicts the future packets that the network will generate and providing the DT with the predictions it generates from the historical network data can alleviate these problems and enable the integration of DTs into IoT networks. Based on this idea, our study aims to accurately determine the throughput of the simulation which is designed for the forest fire detection network environment with the help of a forecaster. The DT estimates the throughput of the network employing the forecaster's output. Due to the high physical and hardware demands of the network, such as square kilometers of coverage and dozens of sensor devices, this study was conducted by simulation. First, the simulation was run and the generated packets were obtained. Then, the forecaster was trained with the packets generated at a portion of the simulation. Then, these forecasts were forwarded to DT, and the throughput of the network over time was estimated by the DT model. To evaluate the performance of the system, the actual throughput values obtained from the simulation are compared with the predictions. The throughput of the network is affected by the packets generated and the packet losses in the network. The system needs to understand both of these components accurately in order to make successful predictions. Studies in the literature have shown that successful predictions can be made with techniques like Recurrent Neural Network (RNN) based models in sequence data. In addition, Graph Neural Network (GNN) based DT models in the research can successfully comprehend situations such as traffic and packet loss. Hence, it is thought that the results that will emerge with the cooperation of these models could be successful.

The results obtained show that this system works promisingly, especially for small-scale networks. However, as the number of devices in the network increases, the system's performance decreases critically. This result could be caused by the forecasting error of the whole network, which increases cumulatively with the increasing number of devices.

In the rest of the paper, we first review the literature on the detection and prevention of forest fires from a network perspective and GNN-based DT studies. Then, in Section, the methodology of the study is described. Results and evaluations are given in Section 4. The last section concludes the paper by giving future directions.

2 LITERATURE SURVEY

2.1 Forest Fire Detection Systems

Early detection of forest fires is critical to reducing their damage. However, detecting fires in minutes when they originate in vast forest areas is arduous. To meet this need, forest fire detection systems utilize satellites, UAVs, and sensors. Since it is not feasible to position satellites to continuously monitor the forest, and as UAVs have limited observation areas and need to be recharged for some time, IoT networks have the fastest fire detection capability among these methods.

IoT networks deployed for forest fire detection can be designed in many different ways. First, it is decided how to detect the fire. While fire detection can be done with affordable sensors such as temperature, CO, and humidity, it can also be done with the help of cameras.

In [3], data such as flame, humidity, and temperature were measured and if the designed algorithm detected a fire situation, alarm packets were sent from the nodes to the database server with location information using satellite communication via SAT-202 module.

Next, network topology and communication technologies should be decided. Depending on the frequency of data sent and the network scale, clustered or mesh topologies are often employed. While fault tolerance is higher in mesh networks, clustered topologies provide scalability. In [4], a hierarchical network structure is designed. Two different types of nodes were preferred: central nodes and sensor nodes. Sensor nodes are connected to each other in a tree structure, with the root node being the central node. Zigbee communication is employed in the sensor nodes, the central node also has the cellular network module to send the data to the server. In another study [5], mesh topology is chosen as a network structure. While all nodes are interconnected with LoRa modules, a gateway device sends all generated packets over the internet to the database server. Thanks to Lora's long-distance communication, the authors stated that they could cover an area of 25 square kilometers for less than $5000 with 100 sensor nodes.

In addition, inexpensive options for sensor nodes can be implemented in networks with fixed Cluster Heads, while in mesh networks, all nodes usually have similar capabilities. As mentioned earlier, communication techniques typically used in wildfire detection are expected to support long-distance transmission. However, for networks in a relatively small forest where the detection range of sensors is limited, technologies such as Bluetooth could be preferred. However, this technique would be both expensive and difficult to manage to cover large forest areas. For instance, in [6], ZigBee communication is chosen to transfer packets. Despite a fast 6-minute fire detection, a 560-acre park in the city was covered and a mesh topology was proposed for scenarios with more nodes. Once all decisions have been made, the network is deployed in the forest and the collected data is analyzed. Rule-based fire detection can be done, as well as smart systems that can detect false alarms with Artificial Intelligence (AI) techniques. Although fundamental algorithms that make decisions by checking certain threshold values are sufficient for fire detection, data-driven learning techniques are also popular for systems that can operate with high accuracy with minimal false alarms. In [7] to avoid false alarms, an unsupervised dataset was used to cluster alarms into false and true using the k-means technique. Multiple linear regression models were then trained with these data. In [8], the decision was made by RNN models. The model was trained with the data from sensor nodes and then the incoming data was evaluated with this model and the fire decision was made. Also in [9], a similar study was conducted with ANN models. In this study, instead of two classes such as the presence and absence of fire, different classes such as fire is about to start are also included. Predicting the location of fire spread is also important to reduce its impact. In a study [10], wind sensors and artificial intelligence techniques were used to estimate the area of fire spread.

2.2 GNNs in Computer Networks

The main use of GNNs in computer networks is to model networks with high accuracy. Two different studies compare the performance of GNN-based models with queueing theory modeling [11] [12]. In both studies, GNN-based models significantly outperformed the queueing theory based models.

Thanks to GNNs' real-time and accurate network modeling, many network problems can now be optimized. One area of particular interest is packet routing optimization. In [13], the Deep Reinforcement Learning (DRL) agent optimized packet routing to maximize allocated bandwidth using the network's GNN-based DT. The agent not only outperformed the fluid models but also, unlike these models, was able to adapt to dynamic changes in the network such as link failure, and could be generalized for networks with similar characteristics.

In a similar study [14], a GNN called TwinNet was developed for network optimization. Instead of DRL, a classical optimization algorithm was used to optimize the average per-flow delay. The model worked quite successfully compared to RouteNet and Multi Layer Perceptron (MLP) alternatives. It achieved a Mean Absolute Percentage Error (MAPE) of around 3 percent and an R^2 score of more than

97 percent. In optimization, it has been much more successful than fluid-based models, especially in high-density traffic, since those models cannot model queueing delay.

There are also studies on how to model networks using GNNs. In [15], modular GNN models represented in terms of expressiveness and granularity were designed as xNet. For 3 different usage scenarios, the performance of the proposed model was tested. The model was able to predict delay with a MAPE rate of less than 5 percent for data with sampling intervals of ms.

Network slicing is a technology designed for next-generation network applications where the physical network is divided into virtual networks. GNNs are also utilized in the management of these networks. In [16], a scenario with different delay agreements for different network slices was tested by training a Graph Linformer Network (GLN) based DT with Federated Learning (FL) and using a heuristic optimization method for both delay estimation and meeting these agreements. The model both outperformed state-of-the-art GNNs and was able to meet the Service Level Agreement (SLA) requirements of the optimization algorithm. In [17], end-to-end latency was measured for all slices. It was able to predict DT based on GraphSAGE with less than 5 percent error on all slices. Furthermore, link failures and SLA performance were also tested. In addition, the model was trained for jitter in order to show that DT can be trained faster for a different metric.

GNNs are also used to predict network traffic. In [18], the feature extraction technique was used to derive features from the network data and predict the traffic. Similar performance was achieved with the extracted features and the training time was significantly reduced.

3 OUR WORK

Digital Twin Networks (DTNs) continuously obtain data from the network and predict network parameter(s) according to the data. In this work, topology, traffic, and communication type data are collected from the network, and the throughput of the network is forecasted. On the other hand, in IoT networks, obtaining real-time data from the network is not straightforward. Because IoT networks lack reliable and low-latency communication due to power and budget constraints. Also, it is not possible to detect colluded or interfered packets from received packets at the sink server. The latter problem can be solved by using simulations to gather training data for the DT. For the former one, forecasting the current traffic is proposed. The complete model of the proposed system can be seen in Figure 2.

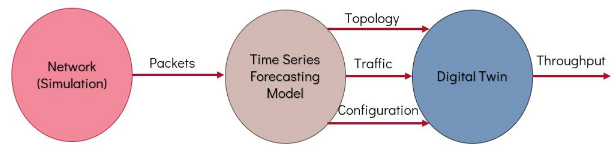


Fig. 2 Model of the System.

3.1 Simulator Design

In order to collect data, a custom WSN simulator is designed since it is easier to reach and tailor the collected data format as needed. To be sure about the reliability of the simulator, its results are compared with the results of the OmNet++ simulator for the selected topology and parameters. Since the results of the two simulators are similar, it is concluded that the designed simulator can be used for dataset generation.

Clustered network topology is preferred for the WSNs in our work. Sensor nodes send the packet to the corresponding cluster head via LoRa-like communication technology, and cluster heads aggregate and forward these packets to the server with a GPRS-like radio modulation technique. These communication technologies are chosen because of their long ranges and relatively low power demands. For the path loss model, the Hata model [19] is employed. Sensor nodes and cluster heads are assumed to be installed at 10 meters in height. They are placed on the trees selected randomly. The server's height is taken as 1000 meters considering variations in the terrain. In the environment, only the thermal noise is calculated, and for simultaneous packets, whether the packet is received successfully is decided based on signal-to-noise and interference ratio (SNIR). Threshold values can be seen in Table 1. The packets of the sensor nodes are considered random events that are generated according to negative exponential distribution. Whereas, cluster heads transmit packets periodically like they are scheduled with TDMA. In case of no received packets, clustered heads may pass their allocated slot without sending a packet. The performance of the model is tested with various number of clustered networks. By changing numbers of clusters, number of sensor nodes, width, and height of the environment, the simulations are repeated. For instance, a 2000x2500 m^2 area is covered with one cluster network, while a 9-cluster network covers an area of 6000x7500 m^2. Topologies used in the study can be seen in Figure 3.

The simulator records all transmitted packets with their status, signal strength, transmission start-end times, and source-destination nodes.

3.2 Forecaster Module

The forecaster predicts whether an individual node is transmitting or sleeping at a given time. As sensor nodes sleep most of the time, the problem that the forecaster addresses can be seen as an imbalanced binary classification problem. Therefore, before training the model, undersampling

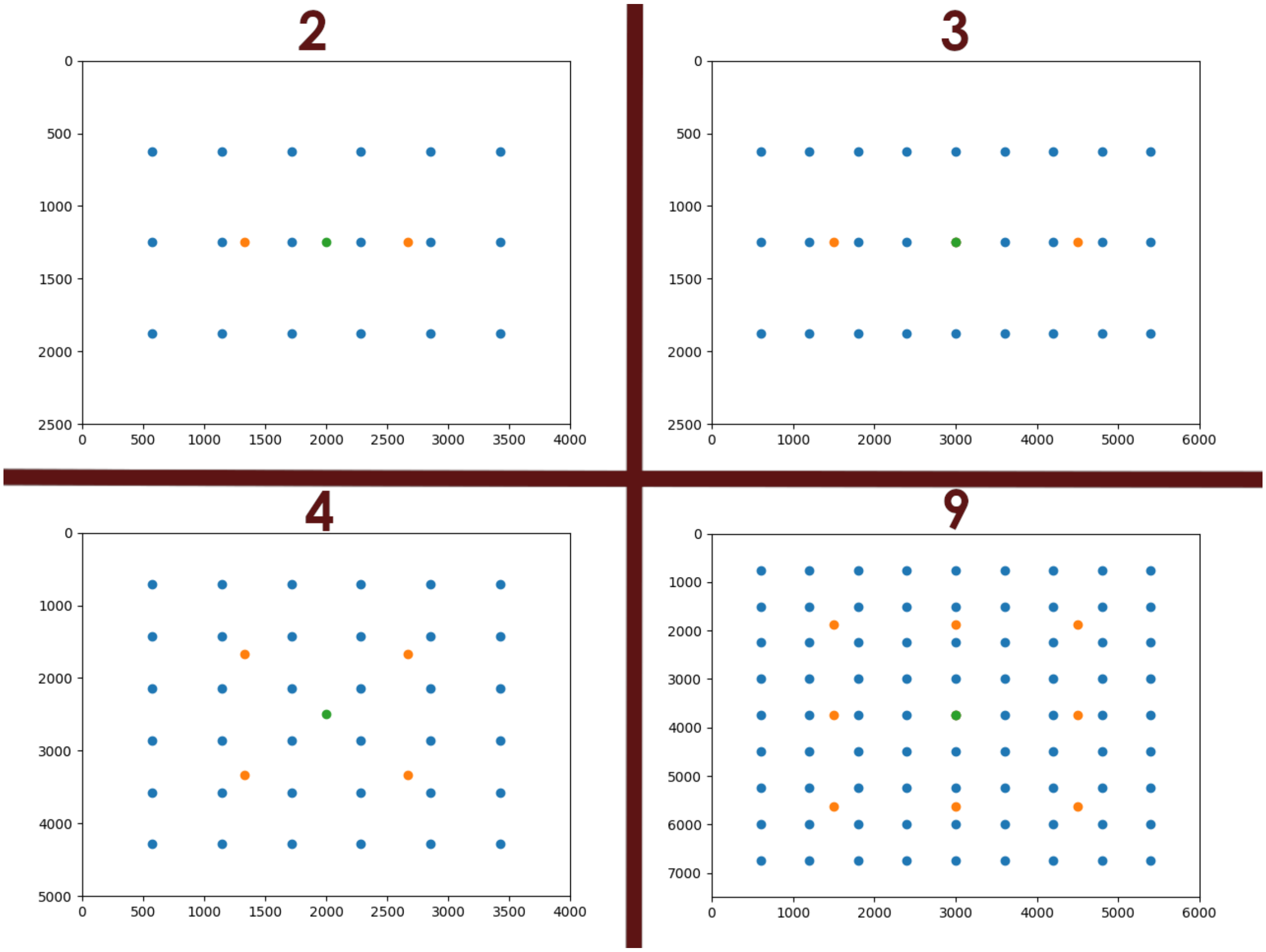


Fig. 3 Various Topologies Tested (having 2, 3, 4, and 9 clusters).

Table 1 Simulation Parameters

Parameter	Value
runtime	24 hrs
packet size	50b
node height	10 meters
GW height	1000 meters
sensor tx power	14 dBm
sensor tx frequency	433 MHz
CH SNIR Threshold	-6 dB
sensor bitrate	5700 Kb/s
mean packet period	1 min
CH min rx power	-130 dBm
GW min rx power	-115 dBm
Temperature	300 K
CH tx power	33 dBm
CH tx frequency	950 MHz
GW SNIR threshold	0 dB
CH bitrate	50000 Kb/s

is applied to the data to prevent bias. Four times as many sleep data samples are randomly selected as the transmitted data samples to train the model. A Long-Short Term Memory-based (LSTM) model is chosen for the forecaster. The model has three LSTM layers and one output fully connected layer to predict the state as 0 (sleep) or 1(transmit). Each LSTM layer has 10 percent dropout rate. From all forecasting results, it is required to retrieve the global state of the network since DT needs it as input. Therefore, after predicting the transmission states of each sensor node and cluster head, the general state of the network including traffic information is constructed. The general state consists of the states of all nodes which are predicted with slight error values. As sleep and transmission states are imbalanced, precision and recall metrics should be considered to decide the performance. F1 score is a metric that combines these two values. The performance of the forecaster is evaluated with the F1 score of the transmit state. Since most of the states are sleep states, the performance of the model for sleep states cannot be trusted.

3.3 GNN Based Digital Twin

Graph Attention Network (GAT) is used in DTs. After the GAT layer, a four-layered multi-layer perceptron (MLP) is placed to predict the throughput. Therefore, DT has one output. The structure of the model can be seen in Figure 4. This network also has 10 percent dropout ratio. ReLu activation function is applied. A considerably large batch size of 256 is used to train. However, the MLP layers have 128 nodes as the hidden dimension number. The last layer has one node for throughput. Other parameters of the DT can be seen in Table 2. For node attributes, the node type, that is sensor node or cluster head, and the node's status is given to the model. Moreover, the same features are provided for edge attributes.

Table 2 GNN Parameters

Parameter	Value
learning rate	0.001
batch size	256
hidden layer dimension	128
dropout rate	0.1
train ratio	0.8
epoch number	50

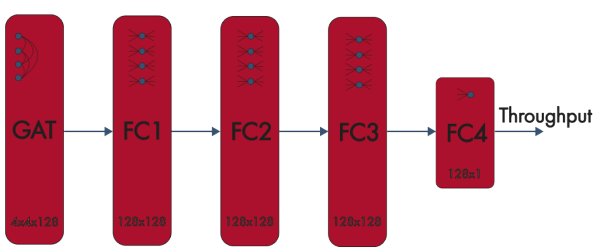


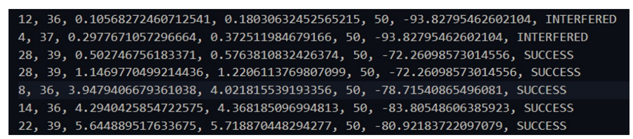
Fig. 4 Structure of the DT.

3.4 How does the System Work?

In the system, traffic was generated in the simulator first. Then, the forecaster was trained by using the a portion of the traffic from the simulator. With, the same packets, DT was also trained. Next, the predictions of the forecaster were fed to the DT to predict the throughput. However, neither the forecaster nor the DT can use the data generated from the previous step directly. Therefore, extra data formation or aggregation steps are added to comply data with the models.

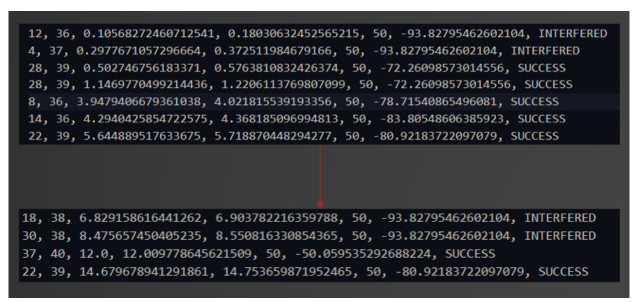
There are five distinct steps of the system. In the first step, packet data is generated with the simulator. The simulator runs the intended simulation and gives reports of generated packets that include: sender and receiver ids, time interval that packet transmits, size of the packet, signal strength and the receive status of the packet, as it can be seen from Fig 5. The forecaster aims to predict the upcoming packets based on the previous transmissions as in Fig 6. Nevertheless, it is a bit challenging to make this prediction from the simulation report. Thus, the problem is divided into simpler problems that are time series forecasting. To do that, for every node, the node's state, which is either sleep or transmit, is inferred throughout the simulation with predefined sampling intervals. This step is named data augmentation and visualized in Fig 7.

After the augmentation, for every node, time series binary classification is done to predict the future states of the nodes as shown in Fig 8. Yet, as the DT requires the total state of the network to predict the throughput, The entire state of the network must be created from the forecasted states of the nodes for each sampled time that throughput is predicted. The network state generation step creates these states for the DT as shown in Fig 9. Then, finally, the DT predicts the throughput of the network for each generated state. Prediction of DT for an example time sample is demonstrated in Fig 10.



```
12, 36, 0.10568272460712541, 0.18030632452565215, 50, -93.82795462602104, INTERFERED
4, 37, 0.2977671057296664, 0.372511984679166, 50, -93.82795462602104, INTERFERED
28, 39, 0.502746756183371, 0.5763810832426374, 50, -72.26098573014556, SUCCESS
28, 39, 1.1469770499214436, 1.2206113769807099, 50, -72.26098573014556, SUCCESS
8, 36, 3.9479406679361038, 4.021815539193356, 50, -78.71540865496081, SUCCESS
14, 36, 4.2940425854722575, 4.368185096994813, 50, -83.80548606385923, SUCCESS
22, 39, 5.644889517633675, 5.718870448294277, 50, -80.92183722097079, SUCCESS
```

Fig. 5 Generated Packet Reports with Simulator.



```
12, 36, 0.10568272460712541, 0.18030632452565215, 50, -93.82795462602104, INTERFERED
4, 37, 0.2977671057296664, 0.372511984679166, 50, -93.82795462602104, INTERFERED
28, 39, 0.502746756183371, 0.5763810832426374, 50, -72.26098573014556, SUCCESS
28, 39, 1.1469770499214436, 1.2206113769807099, 50, -72.26098573014556, SUCCESS
8, 36, 3.9479406679361038, 4.021815539193356, 50, -78.71540865496081, SUCCESS
14, 36, 4.2940425854722575, 4.368185096994813, 50, -83.80548606385923, SUCCESS
22, 39, 5.644889517633675, 5.718870448294277, 50, -80.92183722097079, SUCCESS

18, 38, 6.829158616441262, 6.903782216359788, 50, -93.82795462602104, INTERFERED
30, 38, 8.475657450405235, 8.550816330854365, 50, -93.82795462602104, INTERFERED
37, 40, 12.0, 12.009778645621509, 50, -50.059535292688224, SUCCESS
22, 39, 14.679678941291861, 14.753659871952465, 50, -80.92183722097079, SUCCESS
```

Fig. 6 Expected Forecasting Operation.

4 RESULTS OBTAINED

In this section, first, the forecaster and DT results are evaluated separately. Then, the performance of the whole system is discussed.

4.1 Performance of the Forecaster

The simulator was run for 24 hours of data transmission for each topology setting. Then, the first 80 percent of the data was used to train the forecaster model. After the training, the forecaster predicted the remaining 20 percent. The results were compared with the ground-truth values collected

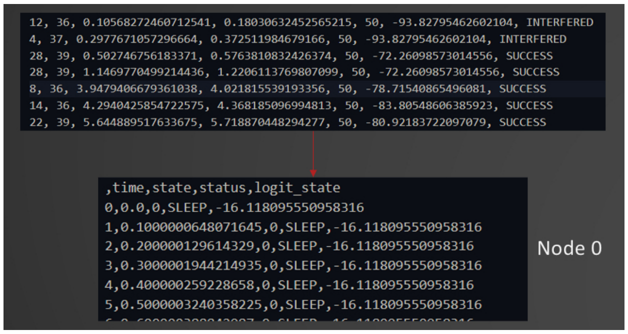


Fig. 7 Data Augmentation for a Node.

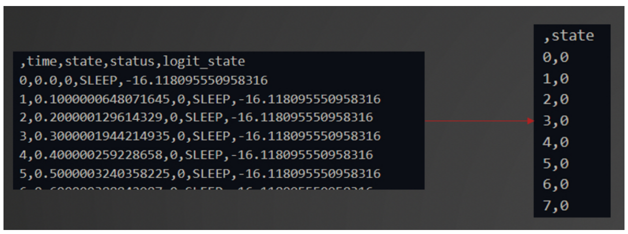


Fig. 8 Forecasting for a Node.

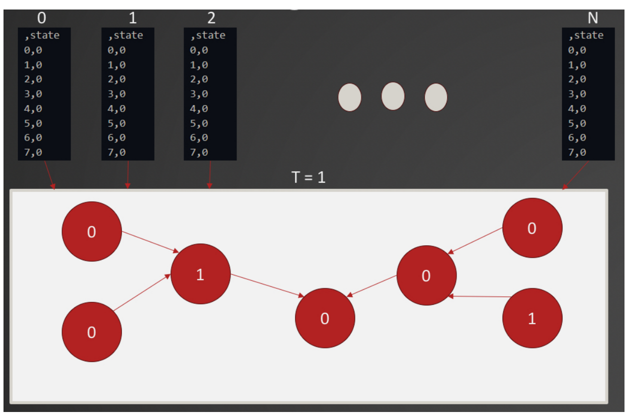


Fig. 9 Network State Generation with Forecasting Results.

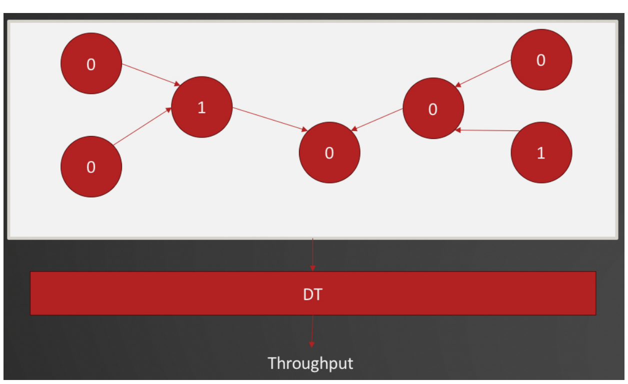


Fig. 10 Throughput Prediction with the DT.

in simulations. The F1 score of the transmission state is calculated to evaluate the model. As forecasting is done for each node separately, the overall result of the forecaster is calculated as the mean and 95 percent confidence interval of all F1 scores of the nodes. The F1 score is calculated as the harmonic mean of the precision and recall values. Since the precision and recall performance of the model is equally important to calculate the throughput accurately, this score is selected as an evaluation metric. It is a prevalent technique for evaluating the performance in imbalanced binary classification problems like the problem that the forecaster solves. Results for the forecaster can be seen in Figure 11. As can be seen, the forecaster can predict the states accurately. Moreover, as the network gets larger, the forecaster's performance is not affected critically. Also, F1 scores of individual nodes in 2-cluster and 3-cluster networks can be observed in Figures 12 and 13, respectively.

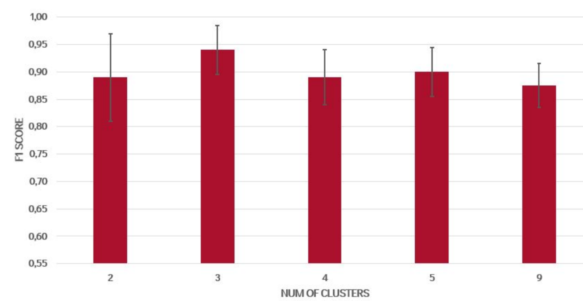


Fig. 11 F1 Scores of Forecaster for Different Topologies.

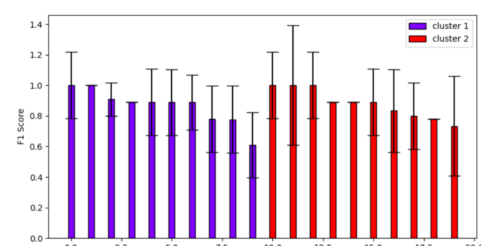


Fig. 12 F1 scores of the nodes in 2 clustered network.

4.2 Performance of the Digital Twin

Ground-truth throughput values were obtained from the simulator. DT also predicted the throughput for test cases following a training. The mean squared error (MSE) and coefficient of determination r-squared (R^2) were used as evaluation criteria for the performance of the DT. The results are given in Figure 14. The GNN-based DT grasps the characteristics of the network as expected. Hence, the R-squared correlation indicator is above 95 percent for all topologies.

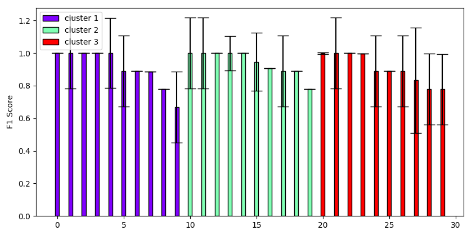


Fig. 13 F1 scores of the nodes in 3 clustered network.

Since the results for all topologies are similar, the scalability of the DT is considered fine.

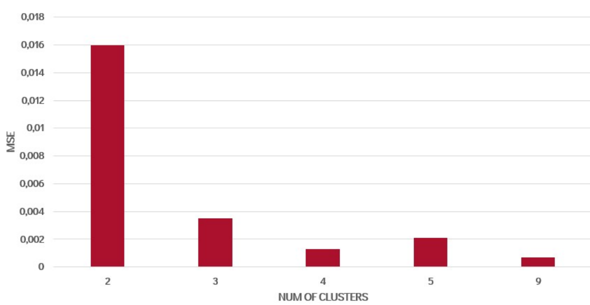


Fig. 14 MSE of the Digital Twin for Different Topologies.

4.3 Performance of the System

To evaluate the integrated performance of the system, the forecaster was run as explained above. In DT, the same training/test split was performed. However, the forecaster's predictions were used for testing instead of the values obtained from the simulator. The same evaluation metrics were used to test the system's performance. Figure 15 gives the results of the whole system. Correlation and MSE results do not vary critically for smaller networks. However, the error increases significantly and the r-squared score cannot show the dependency for 9 clustered networks.

Comparing Figures 14 and 15, it can be seen that DT gives significantly varying results when the actual data and the forecast data are employed. Although GNNs are mostly scalable, the whole system's performance degrades with the increase in the number of nodes here.

The forecaster estimates throughput for each node in the network. Although the prediction performance per node is high, the total error increases as the number of network nodes increases. This causes DT to estimate the throughput with an inaccurate network input. Therefore, the error of the throughput, which is the output of the DT, becomes high. For large networks such as 9 clusters, these estimates are beyond the acceptable limits.

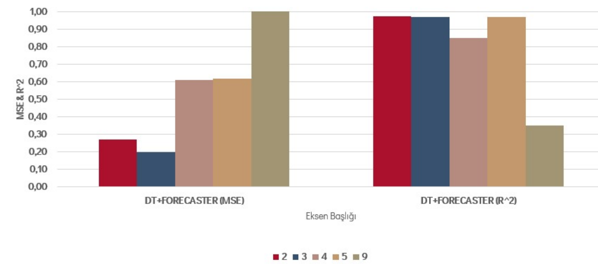


Fig. 15 Performance of the System.

4.4 Effect of Collisions

The results in the previous sections omit the collisions that occurred in the simulations. Although this makes the results unrealistic, the high adaptation and grasping capabilities of GNNs studied by multiple studies in the literature, as well as the system's performance with the collisions, are expected to be similar. To test the performance difference considering collisions, the system was trained and tested under a collision-enabled simulation environment. In order to integrate collisions with the system, collision information was collected from the simulation and ground truth throughput values were calculated accordingly. As the forecaster module predicts the transmissions of the sensor nodes, collision information is irrelevant to the module. Hence, only the DT was re-trained and used the same forecasting states to better determine the effect of collisions. Results can be seen in Figure 16. As expected, the performance of the system is similar when collisions are considered, because of the GNN's understanding of spatial information.

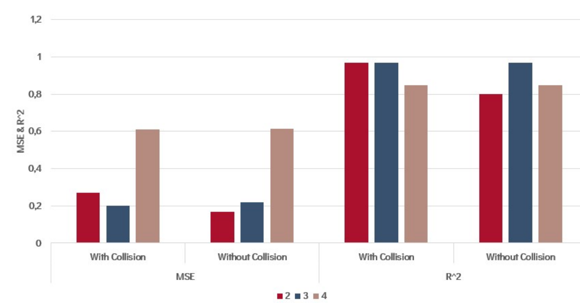


Fig. 16 Collision Effect to the Performance.

5 CONCLUSION

The rise in the average temperature of the world increases the risk of forest fires. Therefore, the importance of forest fire management systems is also increasing. IoT sensor networks are frequently used in these systems to detect forest fires early. However, due to the different requirements of forests, the structures of these networks also vary widely. This requires specific decisions to be made in the management of each network. Due to the size of forests, the management of these large networks can be difficult. DT technology can help network administrators in this area. DTs are widely used in computer networks for optimization and test cases. However, DTs are challenging to use in IoT networks due to their continuous communication requirements. This paper proposes a forecaster based mechanism to facilitate such an integration. The proposed model generates the data needed for DT by predicting the network's packets in advance.

In the study, simulations were performed for networks with different number of clusters, and the generated packets were collected. Then, the forecaster, which was trained with these packets, was asked to forecast the upcoming packets. The DT received the packets generated by the forecaster and was expected to determine the throughput at the given instant. The actual throughput values for this duration were also obtained from the simulation and the performance of the system and the modules were evaluated separately.

In the tests, it was observed that the forecaster module correctly recognized the sent packets with an F1 score of approximately 0.9 for each network type. Moreover, the DT module, when trained independently of the forecaster, achieved an MSE score lower than 0.02 and a high R^2 score of 0.8 in each network. In the test of the network's understanding of collisions, the difference between the collision on and off scores is less than 5 percent. However, when the whole system was integrated, the MSE error increased by more than 100 times and the R^2 score dropped below 40 percent for the network with 9 clusters, although similar scores were obtained for the small-scale networks.

The proposed system seems to have a scalability problem. It is assumed that this is due to the accumulation of errors in the individual predictions of all nodes. Currently, we are working to integrate the other state-of-the-art prediction models to the system. Furthermore, a more holistic forecasting approach and a forecasting model based on clusters can be considered as future work. Also, testing the system under different traffic scenarios and environmental conditions could improve the study.

REFERENCES

[1] OGM. "Official statistics." (accessed: 19/01/2025). (2024), [Online]. Available: https://www.ogm.gov.tr/tr/e-kutuphane/resmi-istatistikler.

[2] V. Chowdary, M. Gupta, and R. Singh, "A review on forest fire detection techniques: A decadal perspective," *International Journal of Engineering & Technology*, vol. 7, p. 1312, Jul. 2018. DOI: 10.14419/ijet.v7i3.12.17876.

[3] K. Grover, D. Kahali, S. Verma, and B. Subramanian, "Wsn-based system for forest fire detection and mitigation," in *Emerging Technologies for Agriculture and Environment*, B. Subramanian, S.-S. Chen, and K. R. Reddy, Eds., Singapore: Springer Singapore, 2020, pp. 249–260, ISBN: 978-981-13-7968-0.

[4] A. Molina-Pico, D. Cuesta-Frau, A. Araujo, J. Alejandre, and A. Rozas, "Forest monitoring and wildland early fire detection by a hierarchical wireless sensor network," *Journal of Sensors*, vol. 2016, no. 1, p. 8 325 845, 2016. DOI: https://doi.org/10.1155/2016/8325845. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1155/2016/8325845. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1155/2016/8325845.

[5] G. Saldamli, S. Deshpande, K. Jawalekar, P. Gholap, L. Tawalbeh, and L. Ertaul, "Wildfire detection using wireless mesh network," in *2019 Fourth International Conference on Fog and Mobile Edge Computing (FMEC)*, 2019, pp. 229–234. DOI: 10.1109/FMEC.2019.8795316.

[6] J. Granda Cantuña, D. Bastidas, S. Solórzano, and J.-M. Clairand, "Design and implementation of a wireless sensor network to detect forest fires," in *2017 Fourth International Conference on eDemocracy & eGovernment (ICEDEG)*, 2017, pp. 15–21. DOI: 10.1109/ICEDEG.2017.7962508.

[7] U. Dampage, L. Bandaranayake, R. Wanasinghe, K. Kottahachchi, and B. Jayasanka, "Forest fire detection system using wireless sensor networks and machine learning," *Scientific Reports*, vol. 12, no. 1, p. 46, Jan. 2022, ISSN: 2045-2322. DOI: 10.1038/s41598-021-03882-9. [Online]. Available: https://doi.org/10.1038/s41598-021-03882-9.

[8] W. Benzekri, A. E. Moussati, O. Moussaoui, and M. Berrajaa, "Early forest fire detection system using wireless sensor network and deep learning," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 5, 2020. DOI: 10.14569/IJACSA.2020.0110564. [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2020.0110564.

[9] P. Pokhrel and H. Soliman, "Advancing early forest fire detection utilizing smart wireless sensor networks," in *Ambient Intelligence*, A. Kameas and K. Stathis, Eds., Cham: Springer International Publishing, 2018, pp. 63–73, ISBN: 978-3-030-03062-9.

[10] Imran, N. Iqbal, S. Ahmad, and D. H. Kim, "Towards mountain fire safety using fire spread predictive analytics and mountain fire containment in iot environment," *Sustainability*, vol. 13, no. 5, 2021, ISSN: 2071-1050. DOI: 10.3390/su13052461. [Online]. Available: https://www.mdpi.com/2071-1050/13/5/2461.

[11] M. Ferriol-Galmés, K. Rusek, J. Suárez-Varela, *et al.*, "Routenet-erlang: A graph neural network for network performance evaluation," in *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*, 2022, pp. 2018–2027. DOI: 10.1109/INFOCOM48880.2022.9796944.

[12] M. Ferriol-Galmés, J. Paillisse, J. Suárez-Varela, *et al.*, "Routenet-fermi: Network modeling with graph neural networks," *IEEE/ACM Transactions on Networking*, vol. 31, no. 6, pp. 3080–3095, 2023. DOI: 10.1109/TNET.2023.3269983.

[13] P. Almasan, J. Suárez-Varela, K. Rusek, P. Barlet-Ros, and A. Cabellos-Aparicio, "Deep reinforcement learning meets graph neural networks: Exploring a routing optimization use case," *Computer Communications*, vol. 196, pp. 184–194, Dec. 2022, ISSN: 0140-3664. DOI: 10.1016/j.comcom.2022.09.029. [Online]. Available: http://dx.doi.org/10.1016/j.comcom.2022.09.029.

[14] M. Ferriol-Galmés, J. Suárez-Varela, J. Paillissé, *et al.*, "Building a digital twin for network optimization using graph neural networks," *Computer Networks*, vol. 217, p. 109 329, 2022, ISSN: 1389-1286. DOI: https://doi.org/10.1016/j.comnet.2022.109329. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128622003681.

[15] M. Wang, L. Hui, Y. Cui, R. Liang, and Z. Liu, "Xnet: Improving expressiveness and granularity for network modeling with graph neural networks," in *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*, 2022, pp. 2028–2037. DOI: 10.1109/INFOCOM48880.2022.9796726.

[16] M. Abdel-Basset, H. Hawash, K. M. Sallam, I. Elgendi, and K. Munasinghe, "Digital twin for optimization of slicing-enabled communication networks: A federated graph learning approach," *IEEE Communications Magazine*, vol. 61, no. 10, pp. 100–106, 2023. DOI: 10.1109/MCOM.003.2200609.

[17] H. Wang, Y. Wu, G. Min, and W. Miao, "A graph neural network-based digital twin for network slicing management," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 1367–1376, 2022. DOI: 10.1109/TII.2020.3047843.

[18] H. Shin, S. Oh, A. Isah, I. Aliyu, J. Park, and J. Kim, "Network traffic prediction model in a data-driven digital twin network architecture," *Electronics*, vol. 12, no. 18, 2023, ISSN: 2079-9292. DOI: 10.3390/electronics12183957. [Online]. Available: https://www.mdpi.com/2079-9292/12/18/3957.

[19] M. Hata, "Empirical formula for propagation loss in land mobile radio services," *IEEE Transactions on Vehicular Technology*, vol. 29, no. 3, pp. 317–325, 1980. DOI: 10.1109/T-VT.1980.23859.