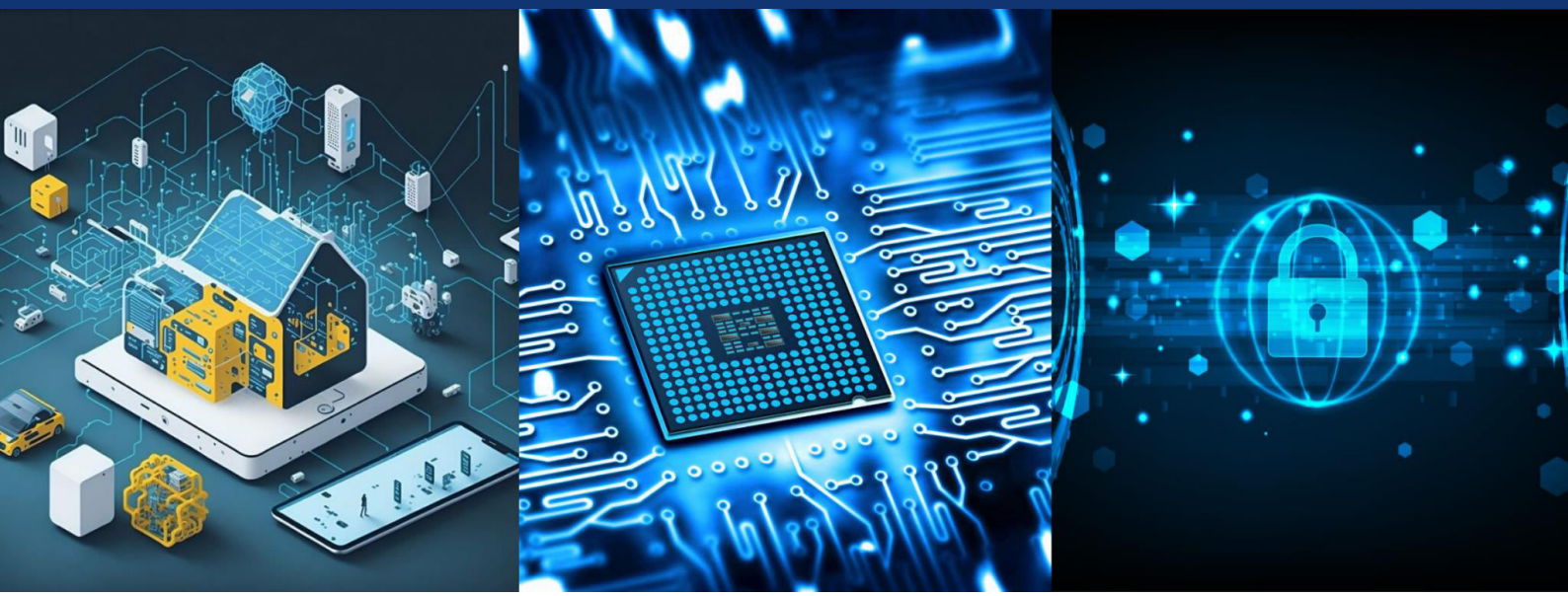
IJIRCCE

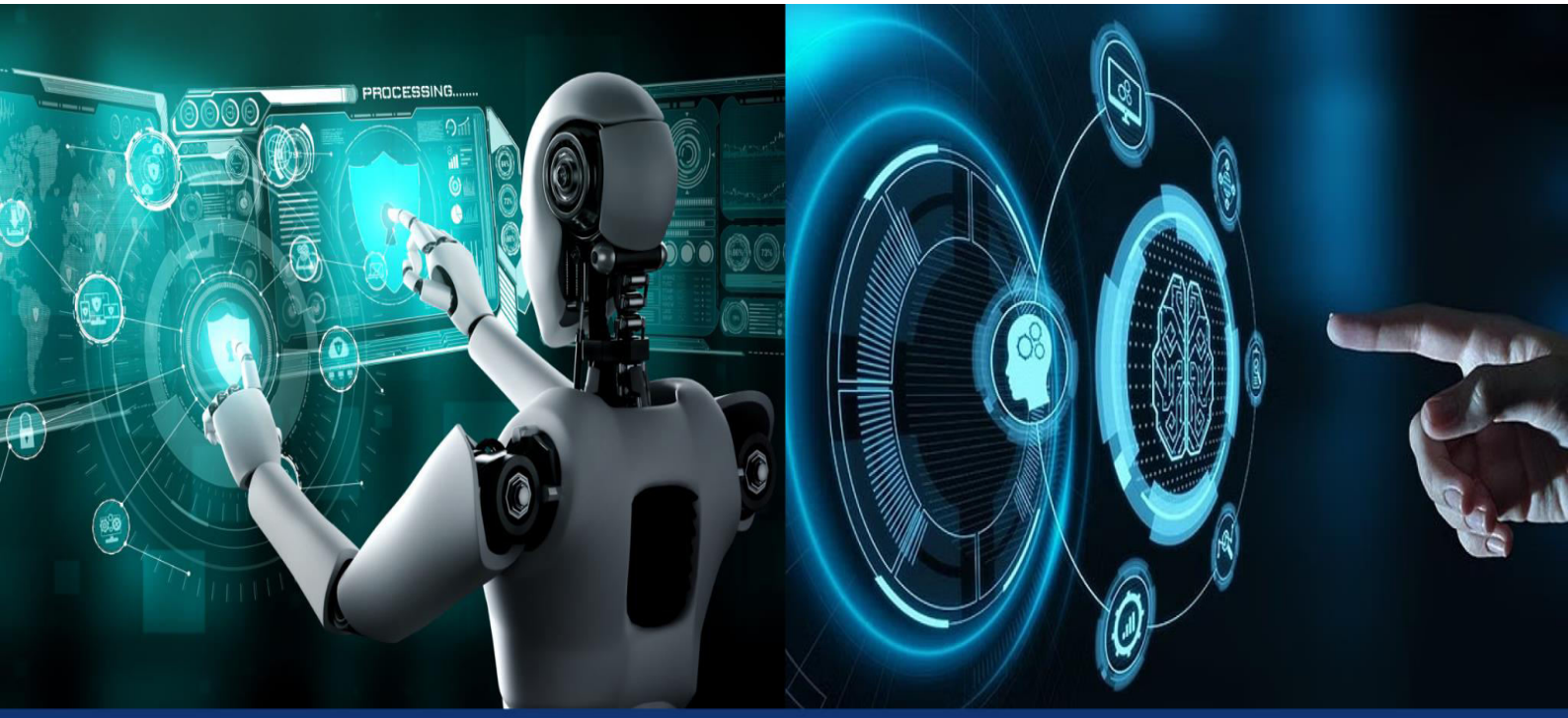
ISSN(O): 2320-9801

ISSN(P): 2320-9798



International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 9.927

Volume 14, Issue 7, July 2026

© 2026 IJIRCCE | Volume 14, Issue 7, July 2026 DOI: 10.15680/IJIRCCE.2026.1407051

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

AI Powered Digital Twin for Self Healing Networks

Gundla Narsimha Raj, Dr. K. Suresh Babu

Post Graduate Student, Department of Computer Science and Engineering, Jawaharlal Nehru Technological University, Hyderabad, India

Professor, Department of Computer Science and Engineering, Jawaharlal Nehru Technological University, Hyderabad, India

ABSTRACT: Modern distributed and edge networks face complexity in fault detection, congestion control, and performance optimization. Traditional systems cannot adapt or automate recovery. AI driven Digital Twins enable real time prediction, simulation, and autonomous fault management. This paper proposes NeuroTwinNet, an AI driven digital twin that mirrors live network states and performs predictive maintenance using AI models. It supports scalable deployment in SDN, IoT, and 5G edge networks, enhancing reliability and resilience. Existing digital twins focus mainly on monitoring or simulation without autonomous correction, and current models cannot adapt to dynamic network behavior in real time. The system uses a Random Forest classifier and a live topology graph to create a feedback loop between the real and virtual network, predicting anomalies, optimizing routing, and recovering nodes autonomously. Experimental results on a Mininet/Ryu based testbed show that the classifier achieves approximately 97% test accuracy and recovers injected link failures in under three seconds with zero observable packet loss.

KEYWORDS: Digital Twin, Self-Healing Networks, Software-Defined Networking, Random Forest, Ryu Controller, Mininet, Reinforcement Learning, and Network Anomaly Detection.

I. INTRODUCTION

Communication networks today carry an enormous and steadily growing share of our daily lives, from online education and remote work to financial transactions, telemedicine, and emergency response. As more services migrate to cloud and edge platforms, the underlying networks have become deeply layered, highly dynamic, and extremely sensitive to even short disruptions. A single failed link in a data centre can cascade into degraded service for thousands of users, and a brief congestion spike can break a real-time call before the operator even sees the alert. Despite this fragility, much of the day-to-day operation of these networks still relies on manual intervention: an engineer reads a dashboard, opens a console, and reconfigures routes by hand. This human-in-the-loop model was acceptable when networks were smaller and changes were rare, but it does not scale to modern conditions.

Software-Defined Networking (SDN) was introduced as a way to break free from this manual model. By separating the control plane from the data plane and exposing a programmable interface to the controller, SDN allows the entire network to be managed and reconfigured through software. The OpenFlow protocol, which is the de-facto standard for SDN, lets a logically centralized controller install forwarding rules on every switch in the network. This central viewpoint makes it possible, in principle, to react to faults in seconds rather than minutes. The remaining question is how the controller should decide what to do, and that is where machine learning enters the picture.

This project, titled "AI Powered Digital Twin for Self Healing Networks", brings SDN and machine learning together in a single working system. A custom Ryu controller monitors switch statistics every few seconds, derives features such as throughput, packet loss, latency, and link utilization, and passes them to a Random Forest model that classifies the network state.

When the model reports congestion or when a link goes down, the controller automatically recomputes the shortest path on a live topology graph using Dijkstra's algorithm, installs new flow rules, and brings the affected traffic back without any operator action. A Streamlit dashboard runs in parallel and shows the live state, the ML prediction, and a timestamped event log so that every healing decision is auditable.

© 2026 IJIRCCE | Volume 14, Issue 7, July 2026 DOI: 10.15680/IJIRCCE.2026.1407051

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

II. RELATED WORK

This work adapts the digital-twin-driven service self-healing philosophy proposed for 6G edge networks, in which a virtual model of the physical network predicts per-flow delay and identifies abnormal links and nodes so that affected services can be redeployed automatically. The present project replaces the graph neural network used in that architecture with a Random Forest classifier and the redeployment algorithm with Dijkstra-based rerouting, adapting the overall philosophy to a Mininet-based SDN testbed. A separate line of work introduces a graph neural network model that predicts delay and other key performance indicators across a wide range of network topologies and traffic configurations, showing that learned models can replace queuing-theory analysis at a fraction of the computational cost. The present project borrows the broader idea that performance metrics derived from raw switch counters can be turned into reliable predictors when paired with the right learning algorithm.

The digital-twin concept has also been discussed more broadly for 5G and future networks, where a continuously updated virtual model of the live infrastructure enables closed-loop optimization, predictive maintenance, and what-if testing of policy changes before they are applied to the real network. The present project adopts a simplified version of this philosophy in which the Ryu controller's in-memory topology graph plays the role of the twin. Foundational work on OpenFlow, which separates the control plane from the data plane and allows a logically centralized controller to install forwarding rules on programmable switches, underlies the entire self-healing mechanism used here, and a comprehensive survey of Software-Defined Networking frames the broader landscape the project sits within, comparing controllers such as Ryu, ONOS, and OpenDaylight and reviewing early work on fault tolerance and recovery.

Despite these advances, most prior work focuses on either prediction or resource optimization rather than closed-loop autonomous recovery. Few studies combine an interpretable ML classifier with a live topology graph and an automated rerouting engine inside a single reproducible testbed. To overcome this limitation, the proposed work introduces an AI-powered digital twin that combines Random Forest-based traffic-state classification with NetworkX-backed Dijkstra rerouting, so that faults are detected and corrected within seconds, without any operator in the loop.

III. SYSTEM ARCHITECTURE AND DESIGN

A. System Architecture:

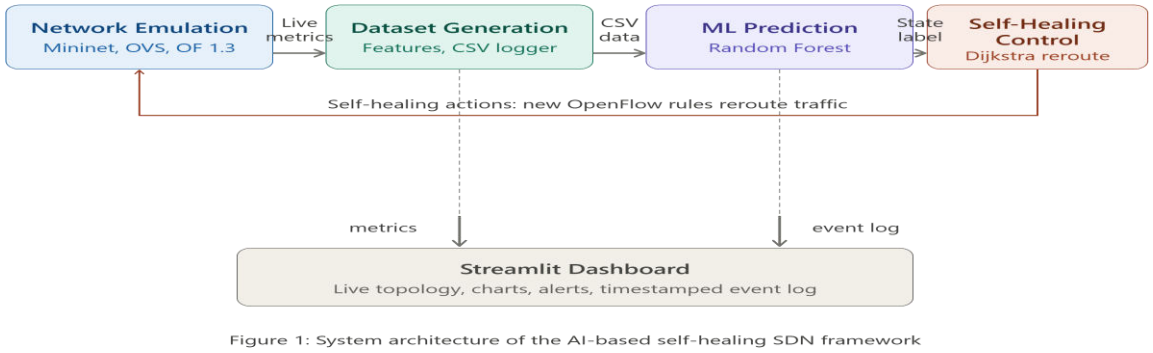


Fig. 1. System Architecture of the AI-Based Self-Healing SDN Framework

B. *Design Considerations:*

- The self-healing framework is trained and tested using labelled traffic generated from a Mininet/Ryu testbed.
- Normal, congestion, and link-failure traffic patterns are all represented in the collected dataset.
- Port statistics are polled every five seconds and converted into throughput, packet loss, latency, and link-utilization features.
- A NetworkX graph mirrors the live topology so that shortest paths can be recomputed the moment a link changes state.
- Random Forest was selected because of its accuracy, interpretability, and suitability for the moderately sized labelled dataset available here.

© 2026 IJIRCCE | Volume 14, Issue 7, July 2026 DOI: 10.15680/IJIRCCE.2026.1407051

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

- Network states are grouped into three categories: Normal operation, Congestion, and Link Failure.
- Only when the classifier reports congestion or link failure is the Dijkstra-based self-healing engine triggered.
- Through a Streamlit dashboard, the system offers real-time state visualization and a timestamped, auditable healing event log.

C. *Description of the Proposed Algorithm:*

In order to preserve network availability and minimize recovery time, the proposed technique aims to precisely classify the live network state and produce an automated corrective action. There are five main steps in the proposed algorithm.

Step 1: Data Collection and Preprocessing:

Port statistics are polled from every switch every five seconds and converted into throughput, packet loss, latency, and link-utilization features. The feature stream is normalized and written to a structured dataset suitable for machine learning analysis.

Let,

$D = \{t1, t2, t3, ..., tn\}$

where D represents the sequence of feature samples (throughput, packet loss, latency, link utilization) collected over n polling intervals.

Step 2: Feature Selection and Model Training:

Throughput, packet loss percentage, link utilization, and latency are used to train the Random Forest classifier. Multiple decision trees are constructed using random subsets of training samples and features.

The final prediction is determined using majority voting:

$$\text{Prediction} = \text{Mode}(T1, T2, T3, ... , Tn) \quad \text{eq. (1)}$$

where T1, T2, ... , Tn represent the predictions generated by individual decision trees.

Step 3: Network State Classification:

The trained Random Forest model analyzes each incoming feature sample and classifies the network state as Normal, Congestion, or Link Failure.

$$\text{State} = RF(F) \quad \text{eq. (2)}$$

where RF represents the Random Forest classifier and F represents the current feature vector.

If State = 0 → Normal Operation

If State = 1 → Congestion Detected Step 4: Affected Link Identification

When congestion or link failure is detected, the controller consults its live NetworkX topology graph to identify the affected switches, links, and flows.

$$P = \text{Dijkstra}(G, src, dst) \quad \text{eq. (3)}$$

where G is the live topology graph and P is the new shortest path recomputed around the affected link. Step 5: Self-Healing Response

After computing the new path, the Self-Healing Engine generates the appropriate flow-rule changes. Healing Action = {Delete Broken Rules, Install New Rules along P}

eq. (4) This closed-loop sequence restores connectivity in under three seconds and requires no operator intervention.

IV. PSEUDO CODE

Step 1: Initialize the Mininet topology and connect all switches to the Ryu controller over OpenFlow 1.3.

Step 2: Begin metric collection on a fixed polling interval by:

- Requesting OFPPortStatsRequest from every connected datapath.
- Computing throughput, packet loss percentage, and link utilization from the raw counters.
- Estimating a smoothed latency value for the interval.
- Appending the timestamped sample to the structured CSV dataset.

Step 3: Separate the collected samples into feature vectors (F) and ground-truth state labels (Y).

Step 4: Split the balanced, augmented dataset (~1300 samples) into training and testing sets.

Step 5: Standardize the features using a Standard Scaler and encode the state labels with a Label Encoder.

Step 6: Initialize the Random Forest classifier with one hundred estimators.

Step 7: Train the Random Forest model on the training set.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Step 8: Evaluate the trained model on the held-out test set.
Step 9: Calculate performance metrics such as Accuracy, Precision, Recall, F1-Score, and five-fold Cross-Validation Accuracy.
Step 10: Persist the trained model, scaler, and label encoder to disk and load them into the live controller.
Step 11: Apply the same scaling and feature transformation steps to each newly collected live sample.
Step 12: Pass the scaled feature vector to the trained Random Forest classifier.
Step 13: Predict the current network state. IF State = Normal
Display "Normal Operation" Continue Monitoring
ELSE
Display "Anomaly Detected" Proceed to Self-Healing
Step 14: Identify the affected switches and links from the live topology graph. IF State = Congestion
Preemptively reroute traffic away from the congested link ELSE IF State = Link Failure
Trigger full self-healing recovery sequence ELSE
Continue routine monitoring
Step 15: Recompute the shortest path using Dijkstra's algorithm on the live topology graph.
Step 16: Perform the flow-rule lifecycle by:

- Deleting the broken flow rules on the affected switches
- Installing fresh flow rules along the newly computed path
- Verifying restored connectivity with a live ping check

Step 17: Display the live topology, metric charts, ML prediction, and event log on the Streamlit dashboard. Step 18: Append the healing decision, old path, and new path to the timestamped JSON event log.
Step 19: Continue monitoring incoming switch statistics for new anomaly events. Step 20: End.

V. SIMULATION RESULTS

The proposed AI-Powered Digital Twin was implemented using Python and evaluated on a Mininet/Ryu testbed deployed on Ubuntu 22.04. The dataset contains labelled samples covering normal, congestion, and link-failure conditions, generated through controlled traffic scenarios and preprocessed for the Random Forest classifier. The model was trained and tested to accurately classify the live network state. Performance evaluation was carried out using standard metrics such as Accuracy, Precision, Recall, and five-fold Cross-Validation. The experimental results indicate that the Random Forest model achieved high classification accuracy while maintaining a fast, reliable detect-decide-act loop, demonstrating its effectiveness in identifying and correcting network anomalies within an SDN environment.

In addition to state classification, the proposed framework incorporates a Dijkstra-based Self-Healing Engine to improve network availability and resilience. Once congestion or a link failure is detected, the system identifies the affected switches and flows and installs a fresh set of flow rules along the recomputed shortest path, deleting the broken rules automatically. The Streamlit-based dashboard provides interactive visualization of live metrics, topology state, ML predictions, and a timestamped healing event log. The overall results demonstrate that the proposed framework effectively enhances service availability, network reliability, and autonomous fault recovery in modern SDN and edge network architectures.

© 2026 IJIRCCE | Volume 14, Issue 7, July 2026 DOI: 10.15680/IJIRCCE.2026.1407051

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

A. Evaluation and Interpretation of System Interfaces:

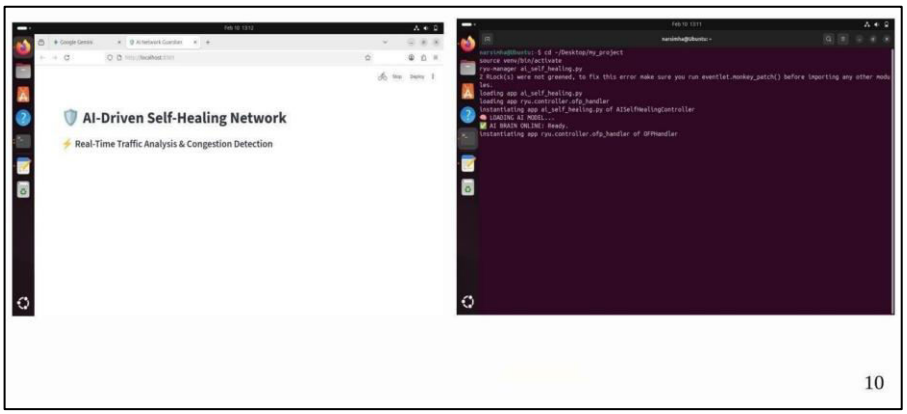


Fig.2. Ryu Controller Launch and Streamlit Dashboard Initialization

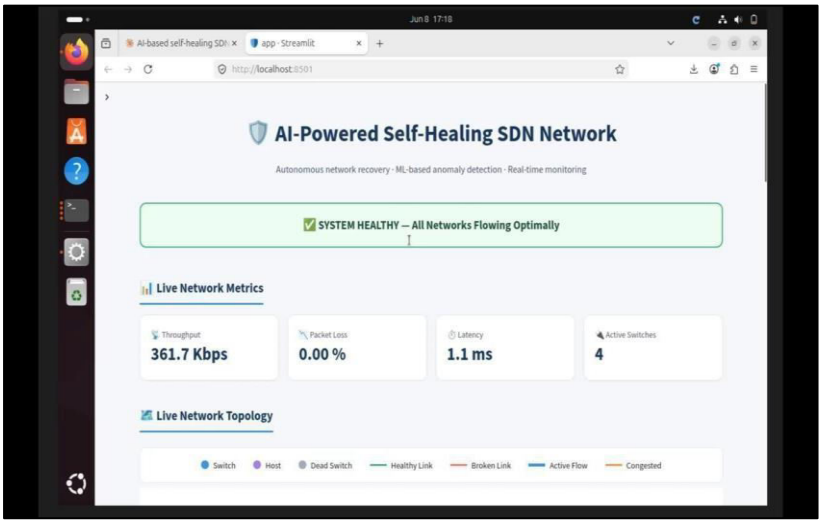


Fig.3. Live Dashboard under Normal Network Operation

Fig.4. Link Failure Detection and Automatic Rerouting

© 2026 IJIRCCE | Volume 14, Issue 7, July 2026 DOI: 10.15680/IJIRCCE.2026.1407051

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

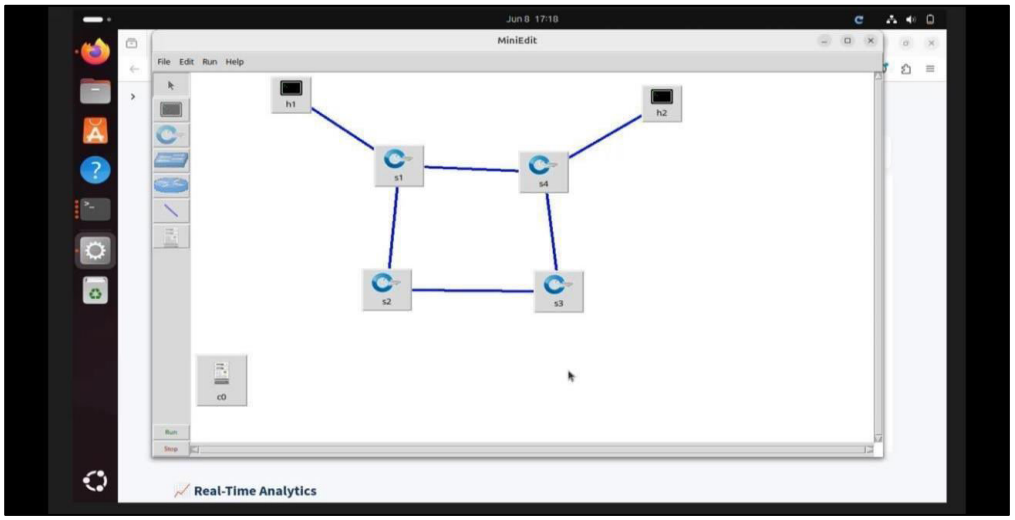


Fig.5. Visual Topology Designed in MiniEdit GUI

B. Metrics for Evaluation:

The performance of the proposed AI-Powered Digital Twin is evaluated using standard classification metrics including Accuracy, Precision, Recall, and F1-Score. These metrics help assess the effectiveness of the Random Forest classifier in distinguishing between the three network states.

Let:

TP = True Positives (correctly identified anomalies, i.e. congestion or link failure)

TN = True Negatives (correctly identified normal operation)

FP = False Positives (normal traffic incorrectly classified as an anomaly) FN = False Negatives (an anomaly incorrectly classified as normal traffic)

Accuracy:

Accuracy represents the proportion of correctly classified network-state samples to the total number of samples analyzed. It measures the overall effectiveness of the model in identifying normal, congestion, and link-failure states.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad \text{eq. (5)}$$

Precision:

Precision is the ratio of correctly predicted anomaly instances to the total number of instances predicted as anomalies.

$$Precision = TP / (TP + FP)$$

eq. (6)

Recall (Detection Rate):

Recall measures the proportion of actual anomaly instances that are correctly detected by the model.

$$Recall = TP / (TP + FN)$$

eq. (7)

F1-Score:

The F1-Score is the harmonic mean of Precision and Recall and provides a balanced evaluation of the classifier's performance.

$$\text{F1Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad \text{eq. (8)}$$

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

VI. CONCLUSION AND FUTURE WORK

This project, AI Powered Digital Twin for Self Healing Networks, effectively classifies live network state using a Random Forest classifier trained on features derived from a Mininet/Ryu SDN testbed, and enhances network resilience through a Dijkstra-based Self-Healing Engine. The system accurately distinguishes normal operation from congestion and link failure, and the generated healing actions restore connectivity within seconds without any operator intervention. Experimental results demonstrate approximately 97% test-set accuracy, 96.4% five-fold cross-validation accuracy, and recovery from injected link failures in under three seconds with zero observable packet loss. Future work includes replacing the Random Forest with a graph neural network trained on a larger dataset, moving the controller to a distributed framework such as ONOS to handle larger topologies, using reinforcement learning to choose among multiple candidate recovery paths, and integrating the framework with real hardware switches through P4-programmable data planes.

REFERENCES

1. P. Yu, J. Zhang, H. Fang, W. Li, L. Feng, F. Zhou, P. Xiao, and S. Guo, "Digital Twin-Driven Service Self-Healing with Graph Neural Networks in 6G Edge Networks," IEEE Journal on Selected Areas in Communications, vol. 41, no. 11, pp. 3607–3623, 2023.
2. M. Ferriol-Galmés, K. Rusek, J. Suárez-Varela, S. Xiao, X. Cheng, P. Almasan, S. Carol, P. Barlet-Ros, and A. Cabellos-Aparicio, "RouteNet-Erlang: A Graph Neural Network for Network Performance Evaluation," in Proc. IEEE INFOCOM Conf. on Computer Communications, pp. 2018–2027, 2022.
3. H. X. Nguyen, R. Trestian, D. To, and M. Tatipamula, "Digital Twin for 5G and Beyond," IEEE Communications Magazine, vol. 59, no. 12, pp. 74–80, 2021.
4. N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling Innovation in Campus Networks," ACM SIGCOMM Computer Communication Review, vol. 38, no. 2, pp. 69–74, 2008.
5. D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey," Proceedings of the IEEE, vol. 103, no. 1, pp. 14–76, 2015.
6. L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
7. E. W. Dijkstra, "A Note on Two Problems in Connexion with Graphs," Numerische Mathematik, vol. 1, no. 1, pp. 269–271, 1959.
8. K. Rusek, J. Suárez-Varela, P. Almasan, P. Barlet-Ros, and A. Cabellos-Aparicio, "RouteNet: Leveraging Graph Neural Networks for Network Modeling and Optimization in SDN," IEEE Journal on Selected Areas in Communications, vol. 38, no. 10, pp. 2260–2270, 2020.
9. L. Ochoa-Aday, C. Cervelló-Pastor, and A. Fernández-Fernández, "Self-Healing and SDN: Bridging the Gap," Digital Communications and Networks, vol. 6, no. 3, pp. 354–368, 2020.
10. Mininet Project, "Mininet: An Instant Virtual Network on Your Laptop (or Other PC)," available online at http://mininet.org, accessed 2026.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details