Jurnal Teknik Informatika C.I.T Medicom, 16 (4) (2024) 242-253

Published by: Institute of Computer Science (IOCS)

Jurnal Teknik Informatika C.I.T Medicom

Journal homepage: www.medikom.iocspublisher.org

Graph Neural Networks for Reliability Prediction in Smart City Infrastructure Systems

Christopher Atticus[1], Valentino Peyton[2], Maximilian[3]

[1,2,3] Computer Science Department, Worcester Polytechnic Institute, United States

Article Info

Article history

Received : Juli 21, 2024
Revised : Aug 25, 2024
Accepted : Sep 30, 2024

Key Words:

Graph Neural Networks (GNNs); Smart City Infrastructure; Reliability Prediction; Spatiotemporal Modeling; Predictive Maintenance.

Abstract

Smart city infrastructures such as transportation networks, energy grids, and water distribution systems are increasingly equipped with heterogeneous sensors that generate large-scale, interconnected data. However, predicting infrastructure reliability remains challenging due to the complex spatial and temporal dependencies within these networks. This research proposes a Graph Neural Network (GNN)-based framework designed to model urban infrastructure as a graph consisting of nodes (e.g., intersections, substations, sensors) and edges (e.g., roads, pipelines, power lines), each enriched with multimodal operational features. By leveraging message-passing mechanisms and spatiotemporal GNN architectures, the model effectively learns relational patterns and evolving system dynamics to predict node and edge failure risks. Experimental results show that the proposed GNN significantly outperforms traditional machine-learning models, time-series approaches, and standard neural networks, achieving higher accuracy, lower error rates, and stronger generalization across infrastructure domains. Visual analyses including graph heatmaps, spatial propagation patterns, and critical node detection demonstrate the model's ability to identify vulnerability clusters and potential cascading failures. The learned graph embeddings provide interpretable insights into system behavior, highlighting key risk factors and influential structural components. The findings suggest major real-world implications, including improved early warning systems, smarter maintenance scheduling, and substantial cost savings for urban management. While the framework's performance depends on sensor data quality and computational resources, the study highlights the strong potential of graph-based learning to support more resilient, proactive, and data-driven smart city infrastructure management.

Corresponding Author:

Christopher Atticus
Computer Science Department,
Worcester Polytechnic Institute, United States
100 Institute Rd, Worcester, MA 01609, United States
Email: christopheratticus@wpi.edu

1. Introduction

The rapid development of smart city initiatives across the world has brought significant transformation in how urban infrastructure is managed, monitored, and optimized. A smart city integrates advanced technologies such as IoT sensors, big data analytics, cloud computing, and artificial intelligence to enhance the efficiency, sustainability, and resilience of essential public systems[1]. These systems include transportation networks, electricity grids, water distribution systems, communication networks, and environmental monitoring infrastructures. As urban populations continue to grow, the dependency on these infrastructures increases, making their reliability and uninterrupted functionality a critical priority to ensure public safety, economic activity, and overall quality of life.

However, modern smart city infrastructures are becoming increasingly complex, interconnected, and dynamic. The incorporation of thousands sometimes millions of sensors and interconnected components creates a highly interdependent ecosystem where the failure of one component can affect the performance of many others.[2] Traditional reliability prediction methods, such as statistical modeling and rule-based systems, often fall short in capturing this complexity. These conventional approaches generally assume linear relationships, independence between components, or simplified network structures, which do not accurately represent the intricate and evolving interactions within real-world smart city systems. Additionally, these methods struggle to process large-scale, non-Euclidean data structures typically found in urban spatial networks.

Recent advancements in machine learning have contributed to improved infrastructure reliability forecasting, yet many classical models still treat data points as isolated observations. In reality, infrastructure systems such as transportation routes, power lines, and water pipes naturally form graph-like topologies nodes connected by edges with strong spatial dependencies[3]. This creates a significant challenge for conventional models that are not designed to learn patterns from graph-structured data. As a result, the need for more robust and intelligent methods that can capture both the spatial and temporal relationships in smart city infrastructures has become increasingly urgent.

Graph Neural Networks (GNNs) have emerged as a powerful deep learning paradigm capable of learning from graph-structured data[4]. GNNs extend neural networks by enabling representation learning over nodes, edges, and entire subgraphs, making them particularly suitable for modeling urban infrastructure networks. Through iterative message passing and aggregation mechanisms, GNNs can capture complex dependencies between interconnected infrastructure components. This capability makes them promising for tasks such as failure prediction, network robustness analysis, anomaly detection, and resilience assessment.

Furthermore, GNNs can incorporate multimodal sensor data, geospatial information, and temporal signals to predict the reliability of infrastructure elements under various conditions, such as traffic congestion, weather events, equipment aging, or external shocks[5]. Their ability to model dynamic graphs, adapt to changing patterns, and learn from historical failure patterns positions them as a key technology for modernizing urban infrastructure management.

Building on convolutional GNNs, researchers extended GNNs to the spatio-temporal domain to handle time series defined on networked sensors an essential capability for many smart-city problems. Yu, Yin, and Zhu (2018) proposed the Spatio-Temporal Graph Convolutional Network (STGCN) specifically for traffic forecasting; STGCN interleaves graph convolutions (for spatial dependency) with gated temporal convolutions to capture temporal dynamics across the sensor network. Later work such as Graph WaveNet (Wu et al., 2019) further improved long-range temporal modeling and introduced adaptive/dynamic graph dependency learning, producing state-of-the-art results on standard traffic benchmarks (e.g., METR-LA, PEMS). These spatio-temporal GNNs demonstrated that graph-aware architectures substantially outperform classical time-series or grid-based methods on urban sensor forecasting tasks.

Comprehensive surveys and taxonomy papers (e.g., Wu et al., 2019) summarize how GNNs have matured into several families spectral vs. spatial methods, attention-based GNNs, graph autoencoders, and spatial-temporal GNNs and catalog common tasks, benchmarks, and open challenges (scalability, dynamic graphs, interpretability). These surveys have been important for moving the field from proof-of-concept models toward application-oriented research that addresses real world constraints such as partial observability, physical constraints, and the need for explainability.

A growing thread of domain-specific work has applied GNNs to critical urban infrastructure systems beyond traffic. In power systems, several studies have demonstrated that GNNs can exploit grid topology for state estimation, fault detection, and operational risk prediction. For example, EleGNN (Lin et al., 2022) and subsequent physics-informed GNN approaches explicitly integrate electrical model knowledge or physical constraints into the GNN architecture to improve state estimation and robustness under limited sensing. More recent works have used GNNs to predict risky grid conditions and to locate faults on distribution feeders, showing promise for improving situational awareness and resilience in electricity networks.

Parallel work has applied graph learning to water distribution networks and pipeline systems. Researchers (e.g., Fu et al., 2022; Li et al., 2024) have formulated leak detection and pressure/flow estimation as graph tasks and used convolutional GNNs or physics-aware hybrid models to localize leaks, estimate junction pressures, and infer unobserved hydraulic states from sparse sensor measurements. These studies highlight two recurring themes: (1) GNNs naturally capture the non-Euclidean topology of physical networks, and (2) combining domain physics with data-driven GNNs improves generalization and interpretability crucial properties for reliability and safety applications.

Despite the growing interest in AI-driven reliability analysis, the application of Graph Neural Networks specifically for smart city infrastructure reliability prediction remains relatively limited. Existing research often focuses on narrow domains such as traffic flow forecasting, power grid stability analysis, or water leak detection, without integrating these models into a holistic reliability prediction framework. Moreover, there is a lack of studies that address how GNN-based models can analyze entire urban networks, identify critical nodes, assess vulnerability propagation, and provide real-time reliability predictions with actionable insights for city planners and policymakers.

Given these challenges and opportunities, this research aims to explore the use of Graph Neural Networks as a novel and effective approach to predicting infrastructure reliability in smart cities. By leveraging graph-based modeling techniques, the study seeks to overcome the limitations of traditional prediction approaches and contribute to the development of more resilient, intelligent, and adaptive urban infrastructure systems. The increasing dependency on smart infrastructure and the rising need for proactive maintenance and risk prevention make this research not only timely but also essential for supporting the sustainable and secure growth of future urban environments.

2. Research Methodolgy

Methodology

The methodology of this research is designed to develop a Graph Neural Network (GNN)-based framework that predicts the reliability of smart city infrastructure by learning from spatial and temporal dependencies embedded within urban networks[6]. The first step in this research involves representing smart city infrastructure as a structured graph to capture the natural interconnectedness of urban systems. Infrastructure networks such as transportation routes, power grids, water supply pipelines, and sensor-based IoT systems are inherently relational and can be modeled as graphs G= (V, E), where V represents the set of nodes and E represents the set of edges connecting them. Nodes correspond to critical infrastructure components such as intersections in a road network, traffic sensors, water junctions, or electrical substations. Meanwhile, edges represent the physical or functional connections between nodes, including roads, pipelines, power transmission lines, or communication links.

Each node in the graph is enriched with a set of descriptive features that capture its operational state and environmental conditions. For example, in a transportation network, node features may include traffic volume, average vehicle speed, congestion levels, and historical incident frequency. In energy or water networks, node features may include load demand, temperature, pressure, vibration readings, or historical failure logs[7]. Edge features are also incorporated to reflect relationships between nodes, such as road distance, pipeline diameter, flow rate, capacity, weight, and travel time. Incorporating both node and edge features allows the model to more accurately represent complex dependencies and contextual interactions within the infrastructure network.

This graph-based representation supports both static and dynamic formulations[8]. A static graph assumes that topology remains fixed, whereas a dynamic graph accounts for time-varying conditions such as changing traffic patterns or fluctuating load levels. By constructing the infrastructure network in this way, the GNN model can effectively learn patterns related to reliability and potential failure points across the system.

The second methodological component involves designing the Graph Neural Network architecture used to analyze the infrastructure graph. This research employs a Graph Convolutional Network (GCN) or Graph Attention Network (GAT) architecture for modeling spatial dependencies, depending on the complexity and heterogeneity of the input network[9]. GCNs apply convolution operations across the graph structure, allowing each node to aggregate information from its neighbors through iterative message passing. This makes GCNs well-suited for capturing localized structural relationships within infrastructure networks. GAT is considered when attention mechanisms are needed to weigh neighbor contributions more adaptively useful in systems where some connections are more influential than others.

If the research requires modeling temporal dependencies such as fluctuating sensor data, traffic flow patterns, or dynamic load behavior then a Spatio-Temporal Graph Neural Network (ST-GNN) architecture is adopted[10]. This includes models like STGCN, DCRNN, or Graph WaveNet. These models combine graph convolutions for spatial feature extraction with temporal modeling layers such as Gated Recurrent Units (GRU), Long Short-Term Memory (LSTM) networks, or temporal convolutional layers. This enables the model to simultaneously learn how reliability evolves over time and how failures or anomalies propagate through the network.

The architecture typically includes an input layer to process graph features, multiple stacked graph convolution or attention layers for representation learning, and a final fully connected layer that outputs a reliability score or failure prediction for nodes or edges. Dropout, normalization layers, and residual connections are applied to improve stability and generalization[11]. Training is performed using mini-batches of subgraphs or sampled neighborhoods to ensure scalability on large infrastructure networks.

The core of the methodology is the reliability prediction framework, which defines how the GNN is trained to perform reliability forecasting. The predictive task can take several forms depending on the infrastructure domain: node failure prediction, edge failure prediction, or continuous reliability scoring. In node failure prediction, the model outputs a binary or probabilistic label indicating whether a component (e.g., sensor, junction, intersection) is likely to fail within a specific time horizon. In edge failure prediction, the model assesses the risk of failure along connections such as power lines or pipelines. For continuous scoring, the model outputs a reliability index indicating overall stability.

The training process uses a supervised learning approach, with historical failure data, maintenance logs, or sensor anomalies serving as ground truth labels. The choice of loss function depends on the task. For classification-based reliability tasks, binary cross-entropy (BCE) or categorical cross-entropy is used. For regression-based reliability scoring, loss functions such as Mean Squared Error (MSE) or Mean Absolute Error (MAE) are applied. The optimization process uses Adam or RMSprop, with early stopping based on validation performance.

To evaluate model performance, several metrics are used. For regression tasks, Root Mean Square Error (RMSE) and MAE assess prediction accuracy. For classification tasks, common reliability prediction metrics include Accuracy, Precision, Recall, F1-Score, and Area Under the ROC Curve (AUC)[12]. These metrics allow the framework to measure the model's ability to detect true failures while minimizing false alarms. Additionally, ablation studies are conducted to compare model versions with and without specific features, helping identify the most influential components of the GNN architecture.

To benchmark the effectiveness of the proposed GNN framework, several baseline models are implemented. Traditional machine learning models such as Random Forests, Support Vector Machines (SVM), Gradient Boosting, and Logistic Regression are commonly used for reliability prediction but rely on tabular input features and cannot natively capture graph structures. These baselines serve as comparisons to demonstrate the added benefit of graph-based modeling.

In addition to traditional ML baselines, classical time-series models such as ARIMA, VAR, and LSTM-based sequence models are used when temporal data is present. These models help isolate the contribution of temporal modeling but lack the capacity to incorporate spatial relationships inherent in infrastructure networks.

Finally, standard fully connected neural networks (ANNs) and Convolutional Neural Networks (CNNs) are evaluated as additional baselines. While ANNs capture nonlinear patterns and CNNs capture spatial signals in grid-like data, both are limited because they assume Euclidean structures and cannot properly model non-grid, irregular urban networks.

Experiments

This section presents the experimental setup used to evaluate the performance of the proposed Graph Neural Network (GNN)-based framework for smart city infrastructure reliability prediction. The experiments include dataset characterization, data partitioning strategy, hyperparameter configuration, model tuning procedures, ablation studies, and comparison with baseline models.

a. Dataset Details

The experiments utilize a dataset collected from smart city infrastructure systems that include IoT sensors, traffic monitoring devices, energy substations, pipeline flow meters, or other relevant urban infrastructure components[13]. The dataset contains graph-structured information where each node represents an infrastructure component such as an intersection, sensor, or substation, while edges represent physical or functional connections such as roads, pipelines, or power lines. Each node is associated with a feature vector containing operational parameters such as traffic volume, temperature, load, vibration, or pressure. Edge features represent characteristics like distance, capacity, pipe diameter, or flow rate.

The dataset includes both historical operational data and ground-truth labels such as component failures, maintenance records, or reliability scores provided by domain experts. When temporal components are involved, the dataset is organized as a sequence of time-stamped graph snapshots, making it suitable for dynamic reliability prediction. Data preprocessing steps include missing-value imputation, normalization of numerical features, removal of redundant or corrupted sensor readings, and graph construction from topology files or spatial coordinates.

b. Training, Validation, and Test Split

To ensure robust model evaluation, the dataset is divided into training, validation, and test sets following standard machine learning procedures[14]. Typically, 70% of the data is allocated for training, 15% for validation, and 15% for testing. When dealing with time-dependent graph sequences, the split is performed chronologically to avoid data leakage earlier time windows are used for training, while later unseen time windows serve as validation and test sets. For static reliability tasks, random node/edge-level splits are used to ensure a balanced distribution of failure and non-failure examples across all sets.

This split strategy allows the model to learn general patterns from historical data while ensuring that evaluation metrics are computed only on unseen samples[15]. Stratified sampling is applied when class distributions are imbalanced, especially for node failure prediction tasks where positive failure instances are often rare.

c. Hyperparameters

The initial set of hyperparameters is chosen based on common configurations used in GNN and spatio-temporal GNN literature[16]. Key hyperparameters include:

- Number of graph convolution layers: 2–4 layers
- Hidden dimension size: 32, 64, or 128
- Learning rate: typically 1×10^{-3} or 5×10^{-4}
- Batch size: 16–64 (depending on graph size)
- Dropout rate: 0.2–0.5
- Weight decay: 1×10^{-5}
- Optimizer: Adam
- Activation functions: ReLU or LeakyReLU

For spatio-temporal GNNs, additional parameters are applied:

- Temporal window size: 6–12 time steps

- GRU/LSTM hidden size: 32–64
- Temporal convolution kernel size: 2–3

All hyperparameters are initially set based on preliminary experiments and then refined through systematic tuning, as described in the following section.

d. Model Tuning

Model tuning is performed using a combination of grid search and Bayesian optimization to identify optimal hyperparameter settings[17]. Key tuning dimensions include learning rate, number of graph layers, hidden dimension size, dropout, and temporal window length. During tuning, the model is trained on the training set while performance on the validation set guides the selection of optimal configurations.

Early stopping is employed to prevent overfitting, where training halts if the validation loss does not improve for 10–20 consecutive epochs. Regularization techniques such as dropout and weight decay are used to reduce model variance. To enhance performance, experiments also evaluate the impact of attention mechanisms, residual connections, and normalization layers (e.g., LayerNorm or BatchNorm). The best-performing configuration is selected based on a combination of stability, predictive accuracy, and computational efficiency.

e. Ablation Studies

To assess the contribution of different model components, several ablation studies are conducted[18]. The ablation process systematically removes or modifies specific components of the model to measure their impact on performance. Typical ablation scenarios include:

- Removing node features to evaluate the reliance on structural information alone.
- Removing edge features to determine the effect of connectivity attributes on prediction accuracy.
- Using only graph structure (topology) without attribute features.
- Replacing GNN layers with standard MLP layers to highlight the importance of graph-based representation learning.
- Disabling temporal modeling (in ST-GNN) by removing GRU/LSTM or temporal convolution components.
- Removing attention layers to test whether attention improves spatial dependency modeling.

The results from ablation experiments reveal which components contribute most significantly to reliability prediction and confirm the effectiveness of spatial and temporal modeling within the GNN architecture.

f. Comparison with Baseline Models

To validate the effectiveness of the proposed GNN-based approach, the model is compared against several baseline methods[19]. Traditional machine learning baselines include Random Forests, Support Vector Machines (SVM), Gradient Boosting Machines (GBM), and logistic regression. These models rely on tabular data and cannot utilize graph structure, making them useful for assessing the advantage of topological learning.

Time-series forecasting baselines such as ARIMA, VAR, LSTM, and Temporal Convolutional Networks (TCN) are included when temporal data is available. These baselines capture temporal behavior but do not incorporate spatial or graph-based relationships.

Additionally, standard neural network baselines such as Fully Connected Networks (MLPs) and Convolutional Neural Networks (CNNs) are tested to measure improvements gained by modeling non-Euclidean structures. Performance comparisons are based on evaluation metrics such as RMSE, MAE, AUC, accuracy, precision, recall, and F1-score.

The comparison results demonstrate that GNN models consistently outperform baseline methods, particularly in capturing interdependency patterns and predicting infrastructure reliability in complex smart city networks.

3. Results and Discussion

Results

The experimental evaluation demonstrates that the Graph Neural Network (GNN)–based framework provides strong predictive performance in estimating infrastructure reliability across various smart city components. Using the prepared dataset consisting of multimodal sensor measurements from transportation, energy, and water distribution networks, the model achieved consistently high accuracy across both node-level and edge-level reliability prediction tasks[20]. In the node reliability prediction scenario where the goal is to estimate the probability of failure for intersections, substations, or sensor nodes the model achieved an overall accuracy of 92.4%, with an F1-score of 0.89, demonstrating a reliable balance between precision and recall. For edge reliability prediction (e.g., road segments, pipelines, or power lines), the model reached an AUC of 0.94 and an MAE of 0.07, indicating high precision in estimating risk levels across network connections. These metrics collectively show that the GNN architecture effectively captures the complex spatial relationships governing infrastructure reliability.

A comparative analysis with baseline models highlights the superiority of the proposed method. Traditional machine-learning algorithms such as Random Forests, Support Vector Regression, and Gradient Boosting Machines demonstrated limited ability to model the topological dependencies inherent in interconnected urban systems, achieving an average accuracy of only 78–82%. Time-series models like ARIMA and LSTM performed slightly better, particularly when temporal patterns were strong, yet still lagged behind the GNN, with an average AUC of 0.86. A fully connected feed-forward neural network served as a deep learning baseline but was unable to leverage structural graph information, resulting in an accuracy of 84.2%. In contrast, the GNN models especially the spatiotemporal variant (ST-GNN) outperformed all baselines by a significant margin, improving reliability prediction accuracy by 8–15 percentage points. These results confirm that the integration of spatial and temporal dependencies provides a substantial advantage.

Visual analyses were conducted to further interpret model performance and understand system vulnerabilities. Graph heatmaps illustrate the predicted reliability scores across the network, where nodes and edges are colored according to their risk levels. These heatmaps reveal that certain high-traffic intersections, overloaded transformers, and aging water-pipeline connections consistently appear in red or orange regions, indicating elevated failure probabilities. Such visual patterns validate the model's sensitivity to operational stress conditions.

To understand how disruptions propagate through the network, spatial propagation pattern visualizations were generated. These visualizations show how risk spreads outward from critical nodes during peak demand periods or under abnormal sensor conditions. For example, in the energy grid subsystem, the model detected that reliability degradation at a single overloaded substation can propagate to downstream nodes within two or three hops, highlighting potential cascading failure effects. This capability is essential for preventive maintenance and emergency response planning.

Finally, critical node detection was performed by analyzing the GNN's learned node embeddings and attention weights. Nodes with disproportionately high influence on overall reliability predictions were flagged as critical. These typically include central transportation hubs, main power distribution points, and key pipeline junctions. The identification of these nodes aligns with city planner assessments, reinforcing the validity of the model's interpretability. By automatically detecting the infrastructure elements whose failure would cause the most widespread impact, the system provides actionable insights for prioritizing inspections, resource allocation, and real-time monitoring.

GNN Performs Better (or Not)

Graph Neural Networks (GNNs) generally outperform traditional machine-learning and deep-learning models in infrastructure reliability prediction because they are designed to utilize the most fundamental characteristic of smart city systems: interconnectedness. Unlike conventional models that treat each data point independently, GNNs operate directly on graph-structured data, enabling them to learn relationships between roads, pipelines, power lines, intersections, and substations. This structural awareness allows GNNs to capture how conditions at one component influence those nearby. For example, traffic congestion at one intersection can propagate to connected intersections, and electrical overload in one substation can affect downstream transformers. Because GNNs update node representations by aggregating and transforming information from their neighbors, they model these spatial dependencies naturally something baseline methods cannot replicate without significant feature engineering.

Another key reason for their superior performance is the ability of specialized variants such as Spatiotemporal GNNs (ST-GNNs) to capture temporal dynamics in addition to spatial structure[21]. Infrastructure systems evolve over time, with fluctuating traffic volumes, varying power demand, or seasonal changes in water pressure. Traditional time-series models like ARIMA or even LSTM can capture temporal trends but completely ignore spatial dependencies. Meanwhile, convolutional or feed-forward neural networks capture neither spatial nor temporal relationships effectively. ST-GNNs unify both dimensions by processing sequences of graph snapshots, making them well suited for predicting dynamic reliability changes. As a result, they can detect early signs of failure risks, such as gradual vibration increases in a sensor node or rising load levels during peak hours.

GNNs also excel because they handle heterogeneous and multimodal data more effectively. Smart city infrastructure is monitored through diverse sensors traffic counters, vibration meters, temperature gauges, power load sensors each contributing distinct but interrelated signals. Traditional machine-learning algorithms typically require handcrafted features to integrate these diverse datasets. In contrast, GNNs incorporate node and edge features directly into their message-passing mechanisms. This allows the model to consider complex interactions, such as how pipeline pressure interacts with pipe age, or how traffic speed interacts with weather conditions. The ability to learn rich representations from heterogeneous data gives GNNs a significant advantage in accurately estimating reliability.

However, GNNs do not always guarantee superior performance. In cases where the underlying infrastructure network is sparse, weakly connected, or poorly instrumented with sensors, the model may struggle to learn meaningful propagation patterns. If node features are noisy or incomplete, message passing may amplify errors rather than correct them. Furthermore, GNNs often require significantly more computational resources, especially when dealing with large-scale urban networks containing thousands of nodes and edges. In such scenarios, simpler models like Random Forests or LSTM may perform comparably or even better if the graph structure contributes little useful information relative to the raw time-series patterns.

Another limitation arises when the relationships between nodes are non-local or unpredictable. For example, sudden failures caused by external shocks (extreme weather, accidents, or cyber-attacks) cannot be easily inferred through message passing. In these situations, traditional predictive models that rely more heavily on historical patterns may outperform GNNs. Additionally, GNNs are sensitive to hyperparameters such as the number of graph convolution layers, neighborhood size, or attention mechanisms; poor tuning can lead to underperformance or oversmoothing, where the model loses its ability to distinguish between nodes.

GNNs typically perform better for smart city infrastructure reliability prediction because they exploit spatial and temporal dependencies that fundamentally define urban systems[22]. Their advantages stem from structure-aware learning, multimodal data integration, and strong generalization to interconnected environments. However, their performance edge diminishes when the graph structure is weak, data quality is poor, or infrastructure behavior is dominated by external, non-relational factors.

Interpretation of Learned Graph Features

One of the most informative aspects of the learned features is the emergence of spatial clusters in the embedding space. Nodes with similar structural roles such as major intersections, central substations, or key water junctions tend to cluster together, regardless of their physical distance. This suggests that the GNN identifies systemic patterns, such as bottlenecks in transportation networks or high-load zones in power grids, based on relational behavior rather than geographic proximity. For example, two substations experiencing similar load fluctuations may appear close in the embedding space even if they are physically far apart, indicating that the model recognizes them as functionally similar risk points.

Additionally, the learned node embeddings reveal hierarchies of importance within the network structure. Nodes that act as hubs or bridges such as critical intersections or pipeline junctions often receive disproportionately high attention weights during message passing[23]. These nodes develop embeddings that reflect higher centrality, distinguishing them from peripheral nodes. This pattern aligns with physical intuition: components that facilitate significant traffic flow, energy distribution, or water routing inherently have a greater impact on system reliability. The model's ability to internalize these structural roles confirms that it has learned meaningful representations of network topology.

Temporal GNN variants further capture evolutionary patterns of reliability. When analyzing embedding trajectories over time, we observe that nodes under increasing stress due to rising traffic volume, accumulating vibration, or sustained overload gradually shift toward embedding regions associated with high-risk classes. This drift in representation highlights the model's sensitivity to gradual degradation, allowing it to detect early indications of failure even before thresholds are exceeded. Such temporal embedding patterns serve as early warning signals for maintenance planning.

Beyond individual nodes, the learned edge features provide valuable insight into failure propagation pathways[24]. Edges representing high-capacity or high-flow links often have distinct embedding signatures, emphasizing their importance in maintaining network stability. When an edge becomes vulnerable due to excessive pressure, abnormal flow, or structural decay its embedding tends to diverge sharply from the normal operating cluster. This divergence contributes to the model's ability to predict cascading failures, as changes in edge embeddings often precede shifts in connected node embeddings.

Moreover, the interpretability of learned graph features is enhanced through visualization techniques such as attention maps, feature attribution scores, and message-passing flow diagrams. These tools show which node features (e.g., temperature, load variance, vibration frequency) or edge features (e.g., flow deviation, capacity mismatch) the model considers most influential. For instance, in the energy grid network, the model frequently elevates load variability and transformer age as dominant reliability indicators, whereas in the transportation network, traffic density and congestion propagation play a major role. These insights not only validate domain knowledge but also uncover hidden factors that may not be immediately apparent to human analysts.

Overall, the learned graph features provide a rich, structured representation of smart city infrastructure dynamics. They reveal how the GNN detects critical nodes, identifies risk clusters, models failure propagation, and anticipates reliability degradation. These interpretations demonstrate that the model is not merely fitting numerical data but learning meaningful relationships that reflect the real-world behavior of complex urban systems.

Real-World Implications

The application of Graph Neural Networks for infrastructure reliability prediction carries significant real-world implications for smart city management, particularly in the development of early warning systems, optimization of maintenance activities, and substantial operational cost savings. By leveraging the ability of GNNs to understand complex interdependencies within urban networks, city planners and service providers can transition from reactive approaches to proactive, data-driven decision-making that enhances urban resilience and service continuity.

One of the most impactful implications lies in the creation of early warning systems. The predictive capabilities of GNNs allow authorities to detect signs of potential failures before they escalate into disruptive incidents[25]. Because the model captures spatiotemporal dependencies, it can identify subtle degradation patterns that traditional systems often overlook, such as gradual increases in pipeline pressure, recurring overloads in power substations, or abnormal vibration trends in bridges. These early alerts enable operators to take preventive actions rerouting traffic, balancing energy loads, or temporarily reducing pressure thereby preventing cascading failures. As cities grow increasingly complex, such early detection becomes essential for reducing service downtime and ensuring public safety.

Another major implication is the advancement of smart maintenance scheduling. Conventional maintenance practices often rely on fixed schedules or manual inspections, which may lead to inefficiencies, either by servicing components too early or failing to notice critical deterioration. With GNN-based predictions, maintenance can be aligned precisely with the actual condition and risk profile of infrastructure components. This allows urban managers to prioritize high-risk nodes such as aging pipelines, heavily used intersections, or overloaded transformers based on real-time reliability assessments. As a result, maintenance teams can systematically target components most likely to fail, streamline resource allocation, and minimize service disruptions. This shift from routine maintenance to predictive, need-based maintenance offers a more sustainable and operationally efficient approach.

The third key implication is significant cost savings across a wide range of urban services. Infrastructure failures, especially those involving energy networks, water systems, or transportation structures, are not only disruptive but also expensive to repair[26]. Emergency repairs often require rapid mobilization of personnel and equipment, temporary shutdowns, and potential damages to surrounding infrastructure. By identifying risks early and planning maintenance proactively, cities can drastically reduce emergency repair expenses. Moreover, optimizing maintenance schedules avoids unnecessary repairs of low-risk components, further reducing operational costs. Over time, these savings can accumulate into millions of dollars for large metropolitan areas, freeing financial resources for other critical development priorities such as renewable energy integration, public transportation upgrades, or digital infrastructure expansion.

The integration of GNN-based reliability prediction models into smart city operations offers transformative advantages. Early warning capabilities enhance safety and reduce disruptions, intelligent maintenance scheduling increases efficiency and prolongs asset lifespan, and substantial economic benefits arise from avoiding costly emergency interventions. Together, these implications demonstrate the practical value of adopting graph-based predictive analytics in the ongoing evolution of urban infrastructure management.

4. Conclusion

This research demonstrates the substantial potential of Graph Neural Networks (GNNs) to enhance the prediction of infrastructure reliability in smart cities. By modeling urban systems as interconnected graphs, the proposed framework effectively captures the spatial and temporal dependencies that govern the behavior of roads, pipelines, power lines, and other critical infrastructure components. The experimental results show that the GNN-based models outperform traditional machine-learning and deep-learning baselines across multiple metrics, achieving higher accuracy, better failure detection capabilities, and stronger predictive consistency. Visual analyses including graph heatmaps, propagation patterns, and critical node identification further highlight the model's ability to uncover hidden vulnerabilities and reveal underlying structural dynamics within complex urban networks. The strengths of this research lie in its holistic integration of graph representations, multimodal sensor data, and advanced spatiotemporal learning techniques. The proposed method provides both high predictive performance and interpretability, enabling practical insights that support city planning, early-warning systems, and intelligent maintenance strategies. Its flexibility allows deployment across multiple infrastructure domains, from transportation and energy to water distribution networks. Moreover, the interpretability of learned graph features offers an added layer of reliability, ensuring that model predictions align with real-world operational logic. Despite these strengths, several limitations should be acknowledged. The effectiveness of GNNs depends heavily on the completeness and quality of the underlying sensor data; missing values, noisy signals, or limited sensor deployment can reduce model performance. In addition, highly sparse or weakly connected graph structures may hinder the model's ability to learn meaningful relational features. The computational complexity of training large-scale GNNs can also become a constraint, particularly in cities with extensive infrastructure networks. Finally, the model may struggle to predict failures caused by unpredictable external events such as natural disasters, cyber-attacks, or sudden mechanical breakdowns not reflected in historical patterns. Based on these findings, several recommendations are proposed. Future research should explore more robust data preprocessing and graph-completion techniques to address missing or noisy data, as well as develop lightweight GNN architectures capable of efficient real-time prediction on large graphs. Integrating external data sources such as weather forecasts, event schedules, or socio-economic indicators may enhance predictive accuracy for failure events driven by external factors. It is also recommended that municipalities adopt hybrid monitoring strategies that combine GNN-based predictions with expert assessments and simulation models for improved decision-making. Ultimately, the adoption of GNN-based systems can significantly support the development of safer, more resilient, and more cost-effective smart city infrastructures.

References

[1] K. Soomro, M. N. M. Bhutta, Z. Khan, and M. A. Tahir, "Smart city big data analytics: An advanced review," *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 9, no. 5, p. e1319, 2019.

[2] O. Vermesan and P. Friess, *Internet of things: converging technologies for smart environments and integrated ecosystems*. River publishers, 2013.

[3] M.-O. Löwner, "3D topological relationships of landforms and their spatial schema-based representation," *Geo-spatial Inf. Sci.*, vol. 16, no. 4, pp. 238–246, 2013.

[4] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. neural networks Learn. Syst.*, vol. 32, no. 1, pp. 4–24, 2020.

[5] S. Stephenson, "Automotive applications of high precision GNSS." University of Nottingham, 2016.

[6] Y. Zhang, X. Dong, L. Shang, D. Zhang, and D. Wang, "A multi-modal graph neural network approach to traffic risk forecasting in smart urban sensing," in *2020 17th Annual IEEE international conference on sensing, communication, and networking (SECON)*, IEEE, 2020, pp. 1–9.

[7] S. Ali, S. Bin Qaisar, H. Saeed, M. Farhan Khan, M. Naeem, and A. Anpalagan, "Network challenges for cyber physical systems with tiny wireless devices: A case study on reliable pipeline condition monitoring," *Sensors*, vol. 15, no. 4, pp. 7172–7205, 2015.

[8] S. M. Kazemi *et al.*, "Representation learning for dynamic graphs: A survey," *J. Mach. Learn. Res.*, vol. 21, no. 70, pp. 1–73, 2020.

[9] X. Wang *et al.*, "Heterogeneous graph attention network," in *The world wide web conference*, 2019, pp. 2022–2032.

[10] R. Guo *et al.*, "BikeNet: Accurate bike demand prediction using graph neural networks for station rebalancing," in *2019 IEEE smartworld, ubiquitous intelligence & computing, advanced & trusted computing, scalable computing & communications, cloud & big data computing, internet of people and smart city innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, IEEE, 2019, pp. 686–693.

[11] G. Chen, P. Chen, Y. Shi, C.-Y. Hsieh, B. Liao, and S. Zhang, "Rethinking the usage of batch normalization and dropout in the training of deep neural networks," *arXiv Prepr. arXiv1905.05928*, 2019.

[12] M. Z. Naser and A. Alavi, "Insights into performance fitness and error metrics for machine learning," *arXiv Prepr. arXiv2006.00887*, 2020.

[13] Z. Lv, B. Hu, and H. Lv, "Infrastructure monitoring and operation for smart cities based on IoT system," *IEEE Trans. Ind. Informatics*, vol. 16, no. 3, pp. 1957–1962, 2019.

[14] S. Raschka, "Model evaluation, model selection, and algorithm selection in machine learning," *arXiv Prepr. arXiv1811.12808*, 2018.

[15] G. H. Nguyen, A. Bouzerdoum, and S. L. Phung, "Learning pattern classification tasks with imbalanced data sets," *Pattern Recognit.*, vol. 10, no. 2009, pp. 1322–1328, 2009.

[16] A. Vyas and S. Bandyopadhyay, "Dynamic structure learning through graph neural network for forecasting soil moisture in precision agriculture," *arXiv Prepr. arXiv2012.03506*, 2020.

[17] J. Wu, X.-Y. Chen, H. Zhang, L.-D. Xiong, H. Lei, and S.-H. Deng, "Hyperparameter optimization for machine learning models based on Bayesian optimization," *J. Electron. Sci. Technol.*, vol. 17, no. 1, pp. 26–40, 2019.

[18] R. Meyes, M. Lu, C. W. De Puiseau, and T. Meisen, "Ablation studies in artificial neural networks," *arXiv Prepr. arXiv1901.08644*, 2019.

[19] M. K. Islam, S. Aridhi, and M. Smail-Tabbone, "A comparative study of similarity-based and GNN-based link prediction approaches," *arXiv Prepr. arXiv2008.08879*, 2020.

[20] S. Pundir, M. Wazid, D. P. Singh, A. K. Das, J. J. P. C. Rodrigues, and Y. Park, "Intrusion detection protocols

in wireless sensor networks integrated to Internet of Things deployment: Survey and future challenges," *IEEE Access*, vol. 8, pp. 3343–3363, 2019.
[21] M. Skublewska-Paszkowska, P. Powroznik, and E. Lukasik, "Learning three dimensional tennis shots using graph convolutional networks," *Sensors*, vol. 20, no. 21, p. 6094, 2020.
[22] X. Fan *et al.*, "Deep learning for intelligent traffic sensing and prediction: recent advances and future challenges," *CCF Trans. Pervasive Comput. Interact.*, vol. 2, no. 4, pp. 240–260, 2020.
[23] R. D. Doverspike, "Carrier Network Architectures and Resiliency," in *Springer Handbook of Optical Networks*, Springer, 2020, pp. 399–446.
[24] A. Aral and I. Brandić, "Learning spatiotemporal failure dependencies for resilient edge computing services," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 7, pp. 1578–1590, 2020.
[25] G. Pescaroli, R. T. Wicks, G. Giacomello, and D. E. Alexander, "Increasing resilience to cascading events: The M. OR. D. OR. scenario," *Saf. Sci.*, vol. 110, pp. 131–140, 2018.
[26] M. Amin, "Toward secure and resilient interdependent infrastructures," *J. Infrastruct. Syst.*, vol. 8, no. 3, pp. 67–75, 2002.