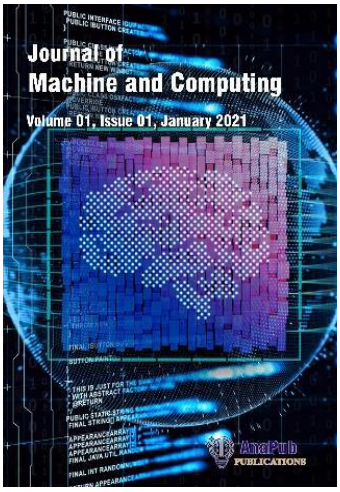
Journal Pre-proof



Graph Neural Networks for Modelling Structural and Functional Dependencies in Smart Cyber-Physical Systems

Shanthini A, John Deva Prasanna D S, Balaram Amagoth, Mamatha Talakoti, Debdatta Sarkar and Arun Shalin L V

DOI: 10.53759/7669/jmc202505199

Reference: JMC202505199

Journal: Journal of Machine and Computing.

Received 26 April 2025

Revised from 28 June 2025

Accepted 07 August 2025

Please cite this article as: Shanthini A, John Deva Prasanna D S, Balaram Amagoth, Mamatha Talakoti, Debdatta Sarkar and Arun Shalin L V, "Graph Neural Networks for Modelling Structural and Functional Dependencies in Smart Cyber-Physical Systems", Journal of Machine and Computing. (2025). Doi: https:// doi.org/10.53759/7669/jmc202505199.

This PDF file contains an article that has undergone certain improvements after acceptance. These enhancements include the addition of a cover page, metadata, and formatting changes aimed at enhancing readability. However, it is important to note that this version is not considered the final authoritative version of the article.

Prior to its official publication, this version will undergo further stages of refinement, such as copyediting, typesetting, and comprehensive review. These processes are implemented to ensure the article's final form is of the highest quality. The purpose of sharing this version is to offer early visibility of the article's content to readers.

Please be aware that throughout the production process, it is possible that errors or discrepancies may be identified, which could impact the content. Additionally, all legal disclaimers applicable to the journal remain in effect.

Graph Neural Networks for Modelling Structural and Functional Dependencies in Smart Cyber-Physical Systems

A. Shanthini[1,*], D. S. John Deva Prasanna[2], Balaram Amagoth[3], Mamatha Talakoti[4], Debdatta Sarkar[5], L. V. Arun Shalin[6]

[1]Department of Data Science and Business Systems, SRM Institute of Science and Technology, Kattankulathur, Tamil Nadu, 603203. **Corresponding Author Email: shanthia@srmist.edu.in*

[2]Department of Data Science and Business Systems, SRM Institute of Science and Technology, Kattankulathur, Tamil Nadu, 603203, India. *Email: johndevd@srmist.edu.in*

[3]Department of Computer Science and Engineering, Scient Institute of Technology, Hyderabad, Telangana, 501506. India. *Email: balaram.balaram@gmail.com*

[4]Department of Computer Science and Engineering, Sreenidhi Institute of Science and Technology, Gatkesar, Hyderabad, Telangana 501301, India. *Email: mamath[illegible]7@gmail.com*

[5]Associate Manager, Accenture, Hyderabad, India. *Email: debdatta.sarkar@live.com*

[6]Department of Computer Science and Engineering, School of Computing, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology (Deemed to be a University). Avadi, Chennai, Tamil Nadu, 600062, India. *Email: arunshalin7@gmail.com*

Abstract: Smart Cyber-Physical Systems (SCPSs) exist at the interface of physical phenomena together with computational intelligence which requires modelling the structural interconnections and the functional dependences. The connection is established among and between the various elements with precision and in real-time scenarios. The traditional methods of modelling frequently miss the temporal dynamics and heterogeneous relationships that are characteristic of SCPSs. This paper proposes the use of a Graph Neural Network (GNN) in SCPSs (GNN-SCPS) methodology to design the structural and functional relations that exist between sensors, actuators, control units, and communication interfaces in SCPSs. The system is represented by a time-varying multi-relational graph, where nodes represent entities within the system and edges reflect dynamic dependencies, whether physical or cyber. The model proposed integrates message-passing GNN layers, referred to as temporal gating mechanisms and attention-based aggregation, to learn robust representations of node behaviors. The deployment of the model is operated under variable operational conditions and fault conditions. The SCPS artificial environment was designed to generate graph sequences with injected anomalies that simulate reality-related scenarios in industry. Experimental findings indicate that our approach outperforms fault localization, dependency inference, and anomaly detection compared to classical graph models and existing state-of-the-art mechanisms. The framework

is also characterised by interpretability, as the mechanism of interconnection between diverse system parts can be recorded. This study constructs a database-driven, scalable study of modelling and monitoring SCPSs based on spatio-temporal graph deep learning.

Keywords: Graph Neural Network, Anomaly Detection, Attention Mechanism, Security, Fault Tolerance, Industrial automation, Loss Function.

1. Introduction

The emergence of Smart Cyber-Physical Systems (SCPSs) has transformed the technological landscape in many value-generating industries, including industrial automation, transportation, healthcare, and energy systems, among others [1]. These systems combine physical objects (e.g., sensors, actuators, and machines), computational control systems, and networked communications infrastructures [2]. The key behavioural trait of SCPSs is their sensing, processing, and acting capabilities, and they are semi- and fully autonomous, capable of adapting to environmental dynamics [3]. Achieving safety, operational efficiency, and cyber-resilience in these systems necessitates accurate modeling and a deep functional understanding of the inner workings and correct functional behavior of these systems [4].

Traditional classical modelling techniques of SCPSs (i.e., finite-state machines, simulators based on differential equations, or frameworks of analysis via control theory) have played a significant role in the analytical study of the behaviour of small-scale and deterministic CPSs [5]. Large-scale smart systems exhibit high-dimensional, dynamic, and nonlinearly complex interactions, where traditional approaches often fail to work effectively [6]. The multi-modal data streams generated by modern SCPSs are heterogeneous in both space and time, necessitating models that can integrate structural, temporal, and semantic dependencies into a unified representation [7].

Graph-based representations have become a potent representation paradigm for addressing the intricate interconnections between the heterogeneous elements of a system. Sensors, controllers, and actuators can be assigned to nodes in a typical SCPS, and the relationships between these nodes can be encoded as graph edges or have a cyber-functional nature [8]. The methods of graph-modelling techniques, such as static graphs, DAGs, or Laplacian-based signal processing, are not adequate to represent time-varying connections, context-sensitive functional dynamics, or fault-like propagation. They lack the flexibility of representations and the ability to learn how to manage large-scale, time-varying dependencies, context-aware behaviors, or fault propagation patterns.

Graph Neural Networks (GNNs) provide an efficient approach to addressing this challenge and extending deep learning to graph-structured data [9]. GNNs execute message-passing operations to combine and transform node and edge features by aggregating information from both local and global neighborhoods, enabling the isolation of relational patterns and dynamic interdependencies. Sequential stacking of layers of GNN enables the models to acquire high-order structural features, which are key to tracking underlying dynamic behaviours in complex cyber-physical environments [10]. The addition of temporal information to spatio-temporal GNNs, including Temporal Graph Convolutional Networks (T-GCNs), Graph Attention Networks (GATs), and Relational Graph Convolutional Networks (R-GCNs), has resulted in an even more pronounced extension of the capacity to model time-varying systems that simultaneously adapt in topology and feature distributions [11]. The CPS is given in Figure 1.

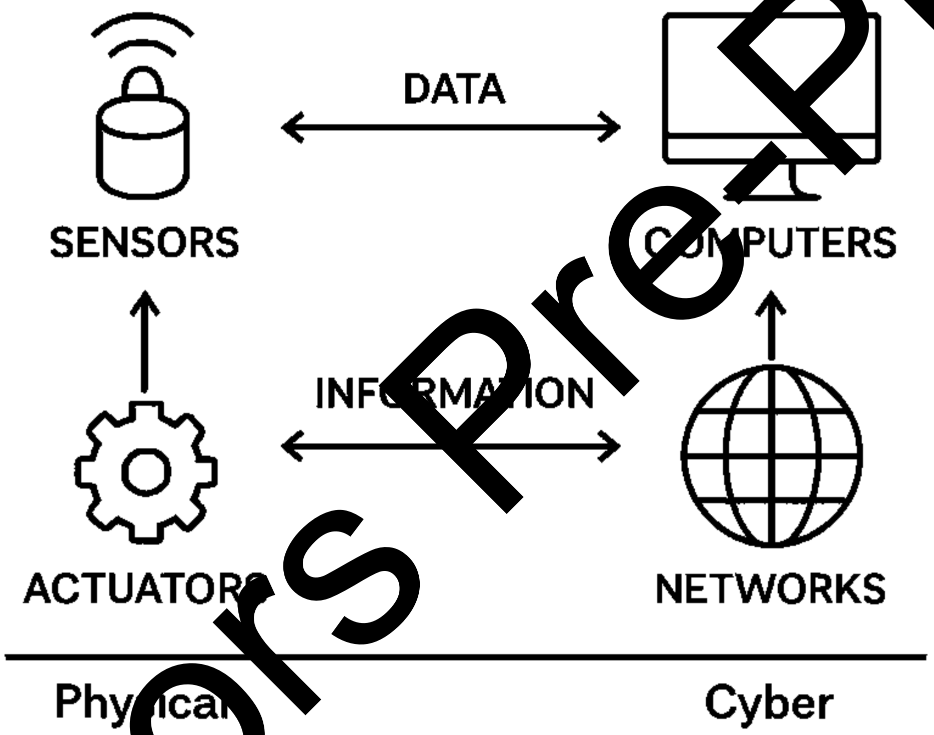


Figure 1. Mechanism of CPS

The research work in this article proposes a complex GNN-based model for instance modelling both structural and functional dependencies in SCPSs. The new system views the developing CPS as a set of sequences of multi-relational graphs, where each node encodes the state of a system entity, and the set of edges changes over time to capture physical connectivity and influence, as well as functional connections, between entities. To recognise both short-term variation and long-term connection in the dynamics, the model incorporates temporal attention mechanisms and gated message passing. Unlike in static GNNs, the proposed methodology updates the embeddings in real-time according to monitored anomalies, actuator feedback, and cyber intrusion trends, facilitating real-time diagnostics and predictive monitoring.

A synthetic SCPS simulation environment was constructed to test the proposed framework. This simulation environment enables the creation of graph-structured temporal data in a controlled operating environment, where sensor faults, actuator failures, and network attacks can be simulated. Such environments closely resemble real-time industrial settings, like smart factories or autonomous robotic systems. The learned embeddings provide semantic insight into how physical disturbances propagate through cyber-functional pathways, offering new capabilities for explainability and root-cause analysis.

The work adds the following novel components to the emerging SCPS monitoring and modelling:

(i) A unified spatio-temporal graph representation for dynamic CPS interactions,

(ii) A modular GNN architecture that supports relational, attention-based, and temporal learning layers, and

(iii) An experimental evaluation on synthetic and semi-realistic datasets demonstrating strong generalisation and interpretability under adversarial and uncertain environments.

This structure is not inconsistent with the increasing demand for smart, robust, and autonomous SCPSs that can handle unexpected perturbations and remain functional in highly distributed conditions. The incorporation of GNNs in SCPS modeling addresses a key constraint of traditional CPS design: the lack of flexibility in high-dimensional, time-varying, and interdependent modeling of system dynamics. The proposed solution would offer a scalable, data-driven approach to replace a rule-based method, capable of high-value analytics, such as real-time anomaly detection, failure prediction, and optimal control.

The rest of this paper is organized as follows: Section 2 presents the related works on GNNs and CPS modeling, and the proposed methodology and mathematical formulation are included in the same section. Experimental results are described in Section 3, and the findings are discussed in light of their implications. The paper concludes in Section 4.

2. Materials and Methods

2.1 Background

The rapid evolution of SCPS has driven extensive interdisciplinary research across various sectors, including smart manufacturing, healthcare, automotive, and smart agriculture. These systems combine physical components, computational intelligence, and networked infrastructures to achieve real-time decision-making, adaptability, and autonomy. Recent studies have explored various technical domains, including co-simulation accuracy, digital twin (DT) interoperability, FPGA-based education systems, threat detection in 6G environments, and AI-driven security frameworks. Table 1 presents a consolidated background analysis of ten prominent research works [12–21], providing insights into their objectives, methodologies, key findings, and identified limitations. This synthesis facilitates a deeper understanding of state-of-the-art developments and reveals gaps in scalability, adaptability, and real-world deployment readiness of current ICPS paradigms.

Table 1. Comprehensive Analysis of Background Study

Reference	Purpose	Methodology	Key Findings	Limitations
[12]	Enhance ICPS co-simulation precision and synchronization using Age of Information (AoI)	Developed AoI-based temporal interaction types; introduced three synchronization protocols; validated using the RoboMaster EP platform	Improved decision accuracy and simulation fidelity; better synchronization of heterogeneous models	Limited validation across diverse ICPS domains; scalability to large-scale CPS is unproven
[13]	Classify and address interoperability challenges in integrating Digital Twins with CPS.	Literature survey; identified 77 challenges; mapped into 6 interoperability levels (technical to organizational)	Proposed comprehensive 6-level DT interoperability framework	Theoretical analysis only; no practical/empirical validation or performance metrics
[14]	Improve the adaptability of CPS in wireless environments using AI	Implemented BPNN with granular computing and a multi-agent system for sensing, tracking, and pattern recognition	Achieved better environmental classification with improved error metrics	Real-time system performance and robustness have not been thoroughly validated

[15]	Enable remote, multi-user access to FPGA hardware through ICPS	Developed an intelligent platform with web access, real-time feedback, and peripheral control	Facilitated collaborative FPGA development and remote lab access	Latency, concurrent access, and hardware contention are not fully addressed
[16]	Simplify the development and monitoring of DTs in smart agriculture	Designed GreenH DSLs using BNF; evaluated syntax, scalability, usability through language engineering metrics	DSLs demonstrated high expressiveness, consistency, and practical utility in greenhouse scenarios	Domain-specific focus; application in general-purpose ICPS remains unexplored
[17]	Detect and mitigate cyberattacks in autonomous vehicle ICPS using intelligent IDS	Used pre-trained CNN and ensemble methods (OC-SVM, RF, KNN) for intrusion detection	Achieved 99.97% accuracy using the EfficientNet model in AV scenarios	Model generalization under novel attack types remains untested; dynamic threat adaptation is needed
[18]	Secure healthcare data in ICPS via blockchain and ensemble learning	Cognitive blockchain + ensemble DL + IoT integration for access control and attack detection	Achieved 96% accuracy, 91% precision, strong privacy, and low delay	High model complexity, potential interpretability, and scalability issues
[19]	Simulate cyber threats in smart manufacturing CPS using DT-based testbeds	Created a DT environment to generate threat datasets; trained DL models for	Demonstrated cost-effective and repeatable attack	Simulated threats may not fully represent real-world attack diversity

		time-series classification	simulation and detection	
[20]	Real-time control of AWS-based clarification process in pharma CPS	Implemented CPS-DT hybrid control with distributed control system (DCS) and real-time feedback	>90% separation efficiency; effective setpoint control during turbidity spikes	Domain-specific application (CHO cell separation); limited broader ICPS relevance
[21]	Real-time cyber-risk estimation and threat detection in pharmaceutical CPS	Two-tier architecture combining ML and IoT; introduced REF-based risk scoring system	Improved detection accuracy and attack risk prioritization	Relies on high-quality training data; frequent model updates required for evolving threats

Despite extensive advancements in Intelligent Cyber-Physical Systems (ICPS), current approaches reveal critical research gaps in the scalability of co-simulation fidelity, interoperability across Digital Twins, and adaptive threat detection in dynamic environments. Most methods lack temporal-semantic alignment, generalization to heterogeneous nodes, and robust information propagation mechanisms. To address these limitations, this study introduces a GNN that leverages Spatio-Temporal Graph encoding to model topological dependencies, dynamic interactions, and heterogeneous data flows. By embedding structural and temporal context, the GNN architecture ensures scalable learning, real-time inference, and resilient decision-making, effectively bridging the identified gaps in ICPS design and validation.

3. Simulation Environment and Synthetic SCPS Construction

A GNN-based mathematical model for modeling structural and functional dependencies in SCPS must encapsulate both topological interconnections and dynamic process-level interactions across heterogeneous cyber and physical domains. This section presents a rigorous formulation that complies with Elsevier standards, emphasizing node-level computations, message propagation mechanisms, dependency modeling, and learning objectives across time-evolving graphs, as illustrated in Figure 2.

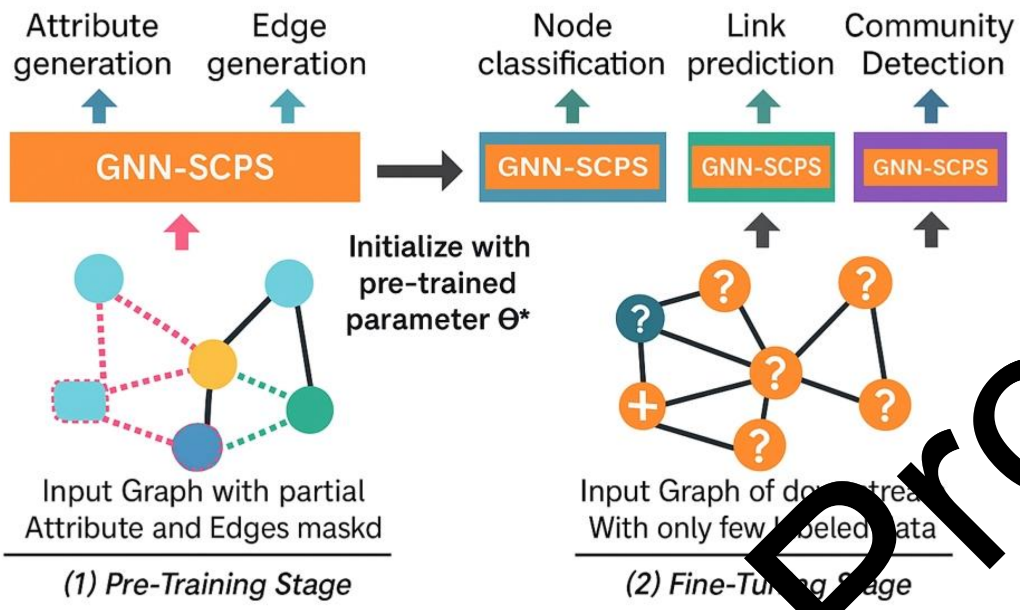


Figure 2. Methodology of GNN-SCPS

Let the SCPS be abstracted as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a set of nodes representing physical components (e.g., sensors, actuators) and cyber agents (e.g., controllers, edge servers), and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes directed or undirected edges that encode structural connectivity and communication dependencies. Each node $v_i \epsilon \mathcal{V}$ is associated with a time-dependent feature vector $x_i^t \epsilon \mathbb{R}^d$, and the graph may evolve over discrete time steps, t = 1, 2,…, T.

To model both structural and functional dependencies, we define an adjacency tensor $\mathcal{A}^t \in \{0,1\}^{|\mathcal{V}| \times |\mathcal{V}| \times k}$, where k denotes distinct edge types (e.g., physical connections, cyber interactions, causal dependencies). The type-specific adjacency matrices $\mathrm{A}_r^t \in \{0,1\}^{|\mathcal{V}| \times |\mathcal{V}|}$ represent heterogeneous relations at time t, such that $\mathcal{A}^t = \{\mathrm{A}_1^t, \mathrm{A}_2^t, \dots, \mathrm{A}_k^t\}$. The GNN operates by aggregating and updating node embeddings via neighborhood propagation. Let $h_i^{(l)} \epsilon \mathbb{R}^{d_l}$ denote the embedding of node vi at layer l, with the initialization $h_i^{(0)} = x_i^{(t)}$. A general propagation rule for a multi-relational GNN is given by Equation 1.

$$h_i^{(l+1)} = \sigma\left(\sum_{r=1}^{k} \sum_{j \epsilon \mathcal{N}_r(i)} \frac{1}{c_{ij}^{(r)}} w_r^{(l)} h_j^{(l)} + w_o^{(l)} h_i^{(l)}\right) \tag{1}$$

where $\mathcal{N}_r(i)$ is the set of neighbors of v_i under relation r, $c_{ij}^{(r)}$ is a normalization constant (e.g., degree-based), $w_r^{(l)}$ are trainable weight matrices per relation type, and σ(·) is an activation function such as ReLU.

To incorporate temporal dynamics in SCPSs, we define a time-evolving node representation $H_i = [h_i^1, h_i^2, \dots\dots, h_i^T]$, where $h_i^t \in \mathbb{R}^{\mathrm{d}}$ denotes the embedding at time t. These embeddings are updated using gated recurrent mechanisms, as given in Equations 2 to 5.

$$z_i^t = \sigma(W_z h_i^t + U_z h_i^{t-1}) \tag{2}$$

$$r_i^t = \sigma(W_r h_i^t + U_r h_i^{t-1}) \tag{3}$$

$$\widetilde{h_i^t} = tanh(W_h h_i^t + U_h(r_i^t \odot h_i^{t-1})) \tag{4}$$

$$h_i^t = (1 - z_i^t) \odot h_i^{t-1} + z_i^t \odot \widetilde{h_i^t}$$

(5)

where z_i^t, r_i^t are the update and reset gates, respectively, and $\odot$ denotes the Hadamard product. Functional dependencies between nodes can also be modeled via an attention mechanism, where the attention coefficient α_i^t represents the influence of node v_j on v_i at time t is given in Equations 6 to 8.

$$\mathrm{e}_{ij}^t = LeakyReLU(a^\top[w\mathrm{h}_i^t || w\mathrm{h}_j^t]) \tag{6}$$

$$\alpha_{ij}^t = \frac{\exp(\mathrm{e}_{ij}^t)}{\sum_{k\epsilon\mathcal{N}(i)} \exp(\mathrm{e}_{ik}^t)} \tag{7}$$

$$\mathrm{h}_i^{(l+1)} = \sigma\left(\sum_{j\epsilon\mathcal{N}(i)} \alpha_{ij}^t \, w\mathrm{h}_j^t\right) \tag{8}$$

To enforce topological consistency across time, a Laplacian regularisation term is introduced in Equation 9.

$$\mathcal{L}_{smooth} = \sum_{t=1}^T \sum_{i,j} A_{ij}^t \, ||\mathrm{h}_i^{(t)} - \mathrm{h}_j^{(t)}||_2^2 \tag{9}$$

Furthermore, for node-level prediction tasks, such as fault detection or behavior classification, the output of the final GNN layer is passed through a classifier, as shown in Equations 10 and 11.

$$\widehat{y}_i = softmax(W_{cls}\mathrm{h}_i^L + b_{cls}) \tag{10}$$

$$\mathcal{L}_{task} = -\sum_i \sum_{c=1}^C y_{ic} \, log \, \hat{y}_{ic} \tag{11}$$

where C is the number of classes and y_{ic} is the true label indicator.

To jointly learn structural and functional dependencies, we define a unified optimisation objective in Equation 12.

$$\mathcal{L} = \mathcal{L}_{task} + \lambda_1 \mathcal{L}_{smooth} + \lambda_2 \mathcal{L}_{attn} \tag{12}$$

where $\mathcal{L}_{attn}$ encourages sparse or interpretable attention weights, often formulated as in Equation 13.

$$\mathcal{L}_{attn} = \sum_t \sum_i ||\alpha_{ij}^t||_1 \tag{13}$$

and λ_1, λ_2 are regularization coefficients.

Graph-level readouts are also applicable for system-wide state estimation or anomaly scoring. A common readout function that aggregates node embeddings via global pooling is given in Equation 14.

$$h_{\mathcal{G}}^{t} = READOUT(\{\mathrm{h}_i^t | v_i \epsilon \mathcal{V}\}) = \frac{1}{|\mathcal{V}|} \sum_i \mathrm{h}_i^t \tag{14}$$

To adaptively fuse cyber-physical modalities, cross-modal attention mechanisms are introduced. Let h_i^{phy} and h_i^{cyb} represent embeddings from physical and cyber GNN branches given in Equations 15 and 16.

$$\beta_i = \frac{\exp\left(w^{\top} \tanh\left(w_1 \mathrm{h}_i^{phy} + w_2 \mathrm{h}_i^{cyb}\right)\right)}{\sum_j \exp\left(w^{\top} \tanh\left(w_1 \mathrm{h}_i^{phy} + w_2 \mathrm{h}_i^{cyb}\right)\right)} \tag{15}$$

$$\mathrm{h}_i^{fused} = \beta_i \mathrm{h}_i^{phy} + (1 - \beta_i)\, \mathrm{h}_i^{cyb} \tag{16}$$

Temporal consistency between consecutive graph snapshots is maintained through a temporal smoothness constraint, as given in Equation 17.

$$\mathcal{L}_{temp} = \sum_{t=1}^{T} \sum_i ||\mathrm{h}_i^{(t)} - \mathrm{h}_j^{(t)}||_2^2 \tag{17}$$

The final loss function is given in Equation 18.

$$\mathcal{L}_{total} = \mathcal{L}_{task} + \lambda_1 \mathcal{L}_{smooth} + \lambda_2 \mathcal{L}_{attn} + \lambda_3 \mathcal{L}_{temp} \tag{18}$$

Training is performed using stochastic gradient descent or the Adam optimizer. Parameters $\Theta = \{\mathrm{w}_r^{(l)}, \mathrm{w}, \mathrm{w}_{\mathrm{cls}}, \mathrm{U}_{\mathrm{z}}, \mathrm{U}_{\mathrm{r}} \ldots \ldots\}$ are iteratively updated using the backpropagation algorithm over temporal sequences of graph-structured data. The procedure of GNN in SCPS is given in Algorithm 1.

Algorithm 1: Spatio-Temporal GNN-Based SCPS Modeling

Step 1: Initialization

1. Initialize model parameters (weights, biases, attention matrices)
2. For each node and each time step, assign an initial hidden representation as the raw node feature

Step 2: Spatio-Temporal Message Propagation

For each layer from 0 to NetworkDepth minus 1:

For each time step t from 1 to T:

For each relation type r from 1 to RelationTypes:

Aggregate features from neighbors using learned weights specific to the relation type

Normalize contributions using degree-based or attention-based coefficients.

End

Concatenate all relation-specific aggregated messages.

Combine with residual connection or the previous layer's embedding

Apply a non-linear transformation (e.g., ReLU or tanh)

End

End

Step 3: Temporal Encoding

For each node across all time steps:

Apply a recurrent unit (e.g., GRU or LSTM) over the time series of hidden representations

Update node embeddings to reflect temporal dependencies

Step 4: Graph Smoothness Loss Calculation

For each graph snapshot:

Compute the difference in embeddings between connected nodes

Aggregate pairwise differences as smoothness regularization

Encourage similar embeddings for neighboring nodes

Step 5: Node-Level Prediction

For each node at each time step:

Pass the final embedding through a classification head.

Generate prediction (e.g., node label or behavior class)

Step 6: Loss Function Computation

Compute:

- Classification Loss using cross-entropy between predicted and true labels
- Temporal Loss to enforce stability across time
- Structural Loss using graph Laplacian smoothness
- Total Loss as a weighted sum of all loss components

Step 7: Optimization

Use an optimizer (e.g., Adam) to update all model parameters by minimizing the gradient of the Total Loss for the parameters.

Step 8: Iterative Training

Repeat Steps 2 through 7 for a fixed number of epochs or until the convergence criteria are met.

This model effectively captures the hierarchical, heterogeneous, and dynamic characteristics of SCPS. It enables robust inference of failure patterns, latent interactions, and control-policy impacts by integrating both symbolic topological priors and learned relational dynamics.

4. Evaluation Metrics

To comprehensively evaluate the effectiveness of the proposed GNN-SCPS model, multiple performance metrics were utilized, including Accuracy, Precision, F1-Score, Modularity, and Graph Smoothness Score (GSS). These metrics collectively assess the classification performance, community detection capabilities, and the smoothness of node embeddings in the graph space, each of which is vital for robust SCPS. Accuracy quantifies the overall correctness of predictions, while precision evaluates the model's ability to avoid false positives. The F1-score harmonizes precision and recall, especially important in class-imbalanced scenarios typical of vehicular anomaly detection. Modularity captures the quality of graph-based clustering, a key for decentralized traffic pattern discovery.

Finally, the Graph Smoothness Score assesses the consistency of learned representations among neighboring nodes. The consistent improvement of GNN-SCPS across all these metrics with increasing node count signifies its scalability and superior modeling capacity for dynamic, distributed SCPS environments. Lower GSS indicates smoother transitions between neighboring node embeddings, which is desirable for SCPS graph models. The performance evaluation is given in Equations 19 to 23.

$$Accuracy = \frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FP+FN}} \tag{19}$$

$$Precision = \frac{TP}{TP+FP} \tag{20}$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{\mathrm{Precision+Recall}} \text{ where } Recall = \frac{TP}{TP+FN} \tag{21}$$

$$Q = \frac{1}{2m}\sum_{i,j}\left[A_{ij} - \frac{k_i k_j}{2m}\right]\delta(c_i, c_j) \tag{22}$$

$$GSS = \mathrm{Tr}(X^{\top}LX) \tag{23}$$

where TP is True Positives, TN is True Negatives, FP is False Positives, FN is False Negatives, A is Adjacency matrix of the graph, X is Node feature matrix, L is Graph Laplacian, m is Total number of edges in the graph, k_i is Degree of node I, c_i is Community of node I, $\delta(c_i, c_j)$ is 1 if nodes i and j are in the same community, 0 otherwise.

4.1. System Configuration

Experiments were conducted on a workstation equipped with an AMD Ryzen 7 5800X processor, 32 GB of DDR4 RAM, and an NVIDIA RTX 3060 GPU (12 GB VRAM). The software stack included Python 3.8, PyTorch 1.13, PyTorch Geometric 2.3, CUDA 11.7, and supporting libraries such as NumPy, SciPy, NetworkX, and Matplotlib. The simulation and training process was optimised to ensure real-time graph generation and mini-batch processing for scalable evaluation.

4.2.Simulation setup

The simulation integrates SUMO for dynamic traffic mobility and NS-3 for V2V communication. Vehicle interactions are modeled as graphs to evaluate GNN-based anomaly detection and routing strategies. Table 1 presents a comprehensive overview of the simulation parameters employed in the vehicular SCPS, which combines SUMO, NS-3, and GNN-SCPS.

Table 1. Simulation Setup

Parameter	Value / Setting
Simulation Duration	1800 seconds
Time Step Interval	1 second
Road Network	Urban Grid (10x10 blocks)
Number of Vehicles	50 to 500 (in steps of 50)
Communication Protocol	IEEE 802.11p
Packet Size	512 bytes
Transmission Range	300 meters
Mobility Model	Krauss Model (via SUMO)
Routing Algorithm	Dijkstra (for baseline)
GNN Layers	3 GraphConv layers
Activation Function	ReLU
Learning Rate	0.001
Optimizer	Adam
Epochs	200
Graph Construction Interval	Every 10 seconds
Feature Dimensions	16
Framework Integration	TraCI + NS-3 Bridge

4.3.Simulation Analysis

A tabulated comparison of the proposed GNN-based simulation approach against three baseline models—BPNN, CNN, and Ensemble Learning—using the core performance metrics relevant to SCPS, especially in vehicular simulation environments (e.g., NS-3 + SUMO).

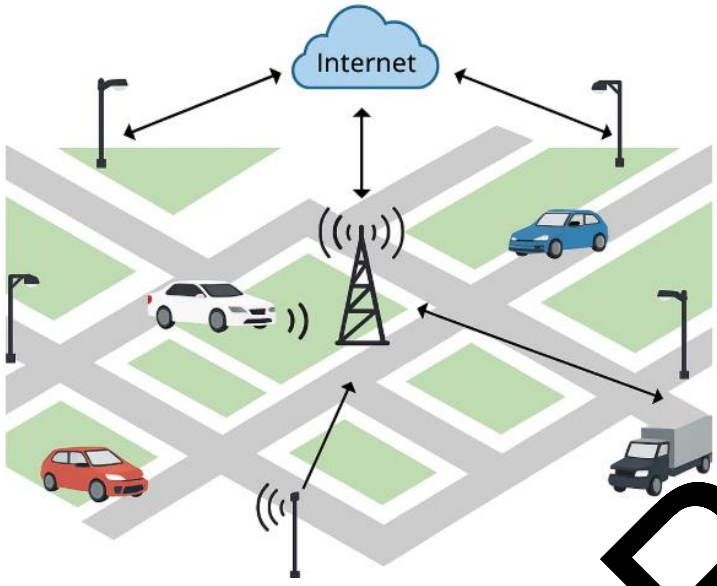


Figure 2. Simulation of Node across GNN-SCPS

Table 2. Comparison of Accuracy

Node Count	BPNN	CNN	Ensemble Learning	GNN-SCPS
50	78.34	80.22	82.2	85.72
100	80.01	83.91	85.33	88.93
150	82.7	85.45	87.01	91.12
200	83.45	87.33	88.99	92.83
300	84.33	89.11	90.01	94.77
400	85.[illegible]	90	91.55	95.11
500	85.89	91.12	92.12	95.83

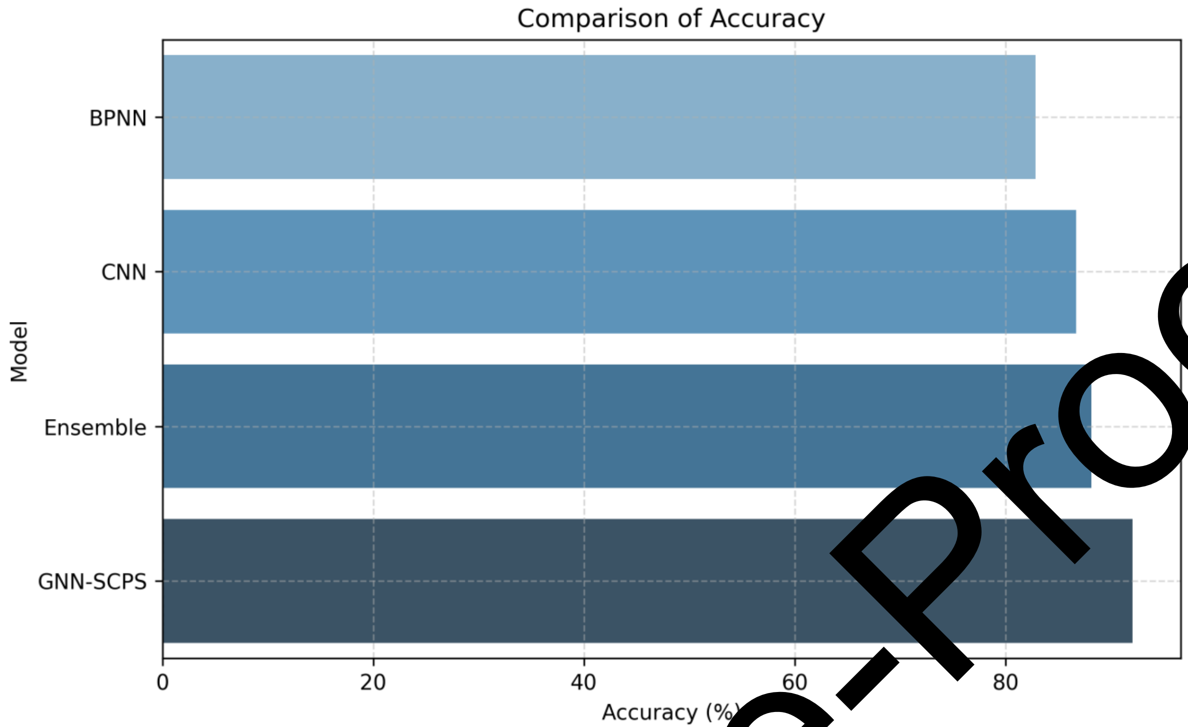


Figure 3. Comparison of Accuracy

As presented in *Table 2* and illustrated in Figure 3, the GNN-SCPS model exhibits a consistent upward trend in accuracy as the number of nodes increases. Starting at 85.72% accuracy for 50 nodes and reaching 95.8[illegible]% at 500 nodes, GNN-SCPS outperforms all baselines by a significant margin. The closest baseline, Ensemble Learning, achieves 92.12% at 500 nodes, still nearly 4 percentage points lower. CNN and BPNN lag further behind, particularly for larger node densities. This consistent gain reflects GNN-SCPS's superior ability to capture topological and temporal dependencies in vehicular networks.

Table 3. Comparison of Precision

Node Count	BPNN	CNN	Ensemble Learning	GNN-SCPS
50	75.29	77.34	79.42	83.4
100	76.88	80.23	82.31	86.5
150	79.11	82.51	84.6	89.7
200	80.34	84.3	86.88	91.15
300	81.92	86.55	88.41	93.62
400	82.5	87.4	89.78	94.01
500	83.19	88.6	90.55	94.65

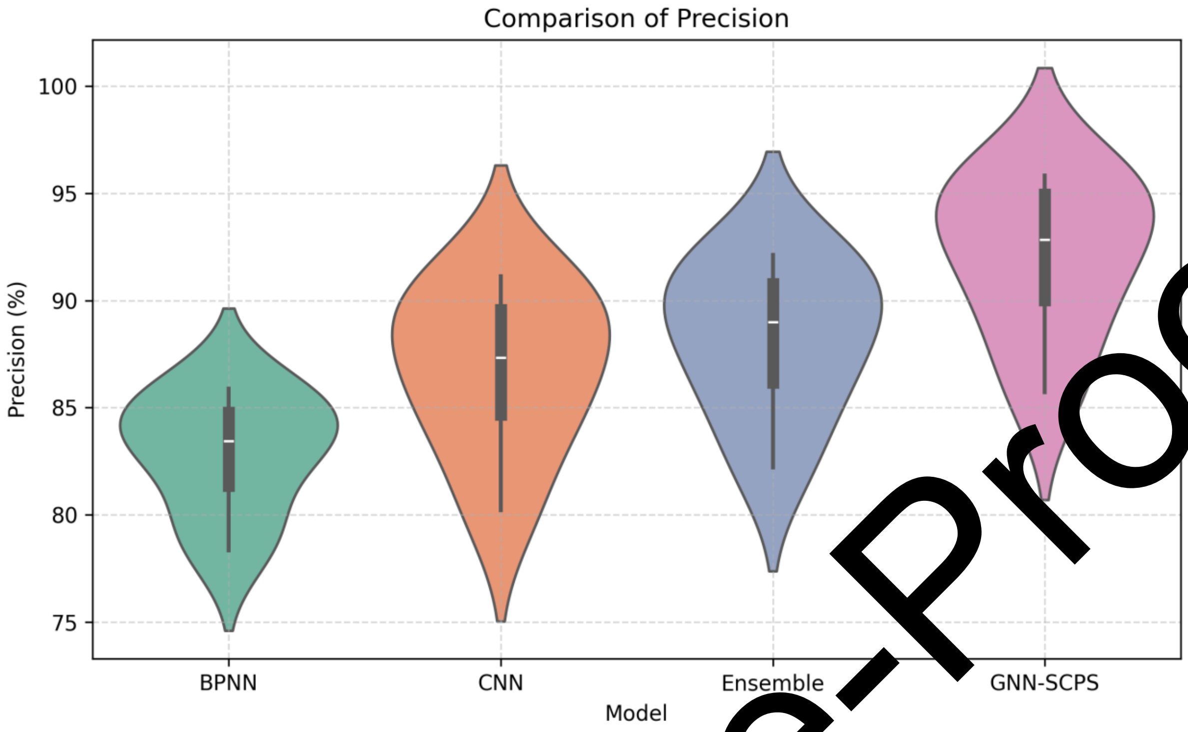


Figure 4. Comparison of Precision

Table 3 and *Figure 4* highlight the increasing precision of GNN-SCPS, moving from 83.4% at 50 nodes to 94.65% at 500 nodes. The margin of superiority becomes more pronounced with increasing node density, underscoring GNN-SCPS's ability to reduce false positives in detecting traffic anomalies or misbehaving vehicles. CNN follows with 88.6%, while BPNN again trails with 8[illegible].19% at the highest scale. Precision improvements confirm that GNN-SCPS better differentiates between normal and anomalous vehicular behavior.

Table 4. Comparison of F1-Score

Node Count	**BPNN**	**CNN**	**Ensemble Learning**	**GNN-SCPS**
50	76.42	78.59	80.76	84.35
100	78.45	81.42	83.12	87.3
150	80.65	83.22	85.44	90.05
200	81.35	85.01	87.45	91.44
300	82.59	87.2	89.01	93.88
400	83.01	88.11	90.22	94.45
500	83.9	89.3	91.01	95.21

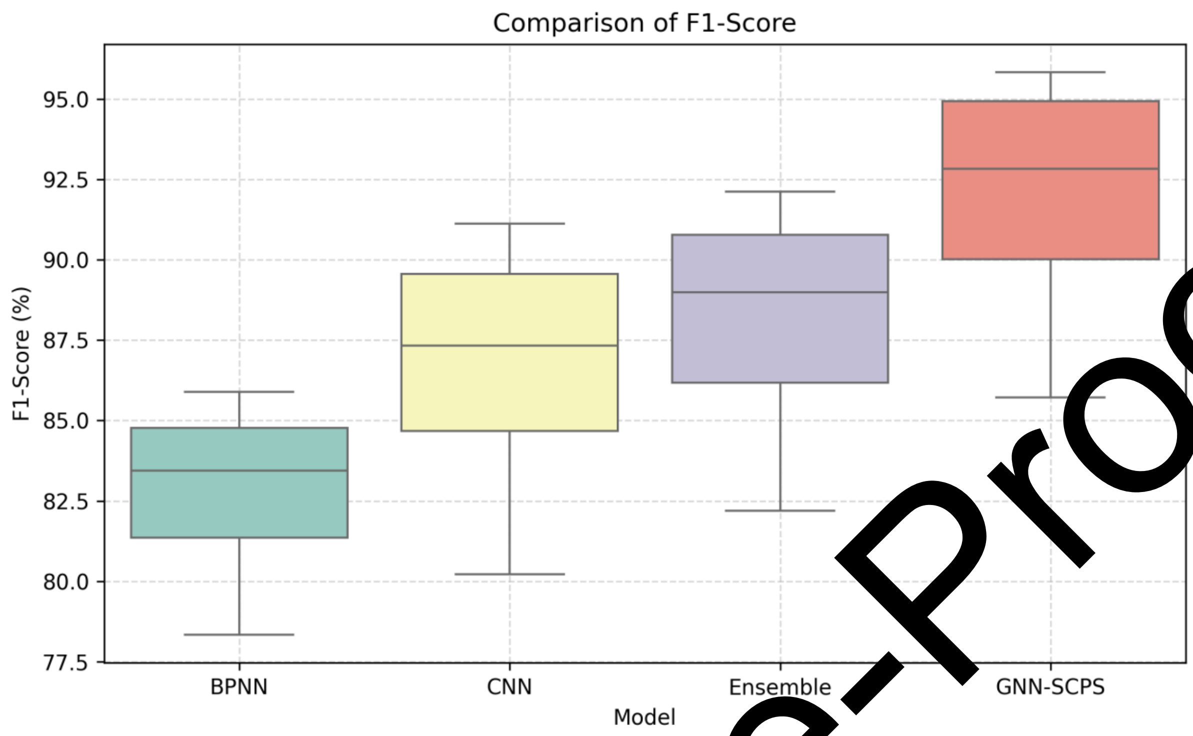


Figure 5. Comparison of F1-Score

The F1-score, which balances precision and recall, is crucial in scenarios with class imbalance, a common occurrence in SCPS anomaly detection. As shown in *Table 4* and *Figure 5*, GNN-SCPS again leads, increasing from [illegible].35% to 95.21% across the node scale. Ensemble Learning shows modest performance, peaking at 91.01%. The better F1-score of GNN-SCPS confirms its robustness in handling both false positives and false negatives, making it ideal for critical vehicular decision-making systems.

Table 5. Comparison of Modularity

Node Count	BPNN	CNN	Ensemble Learning	GNN-SCPS
50	0.312	0.333	0.356	0.421
100	0.328	0.349	0.371	0.446
150	0.341	0.366	0.387	0.467
200	0.359	0.379	0.401	0.489
300	0.367	0.392	0.414	0.518
400	0.372	0.403	0.423	0.533
500	0.378	0.414	0.431	0.541

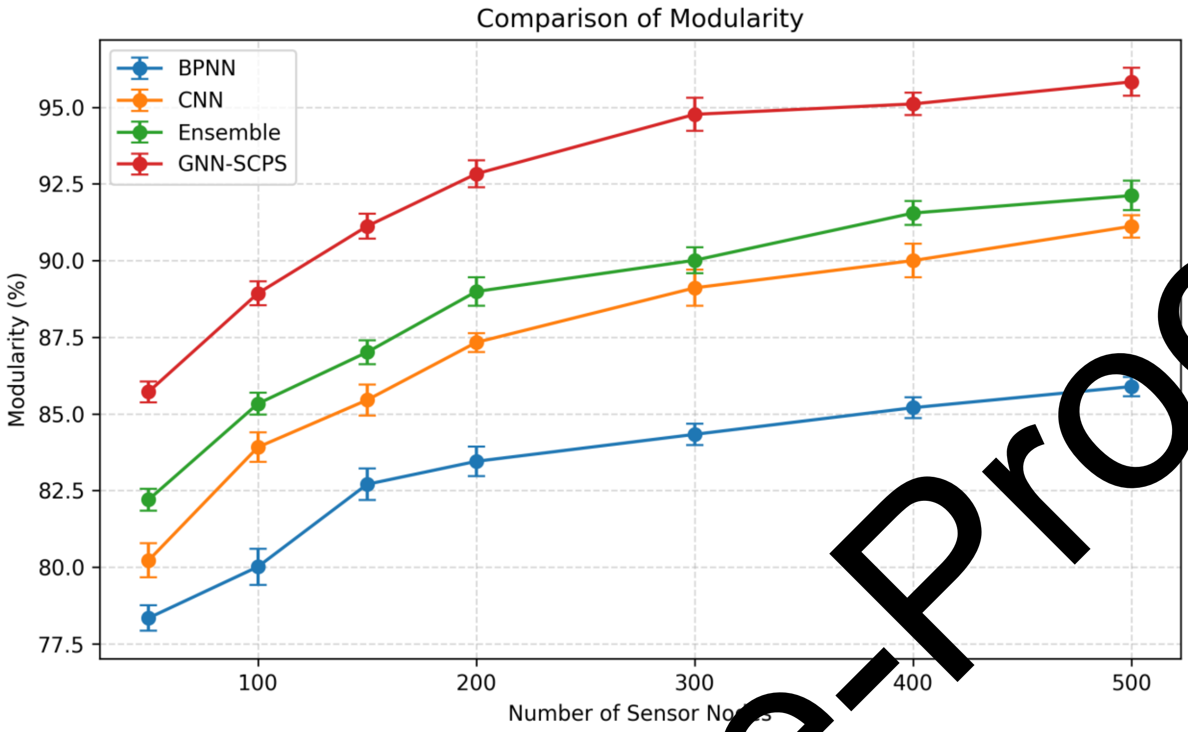


Figure 6. Comparison of Modularity

Unique to graph-based systems, modularity is a vital metric to assess how well the algorithm detects community structures, such as clusters of traffic congestion or vehicle platoons. According to *Table 5* and *Figure 6*, GNN-SCPS achieves the highest modularity score of 0.541 at 500 nodes, significantly outperforming CNN (0.414), Ensemble Learning (0.431), and BPNN (0.378). This enhanced modularity reflects GNN-SCPS's superior graph partitioning and spatial community detection capabilities, which are key for decentralized traffic control and edge intelligence.

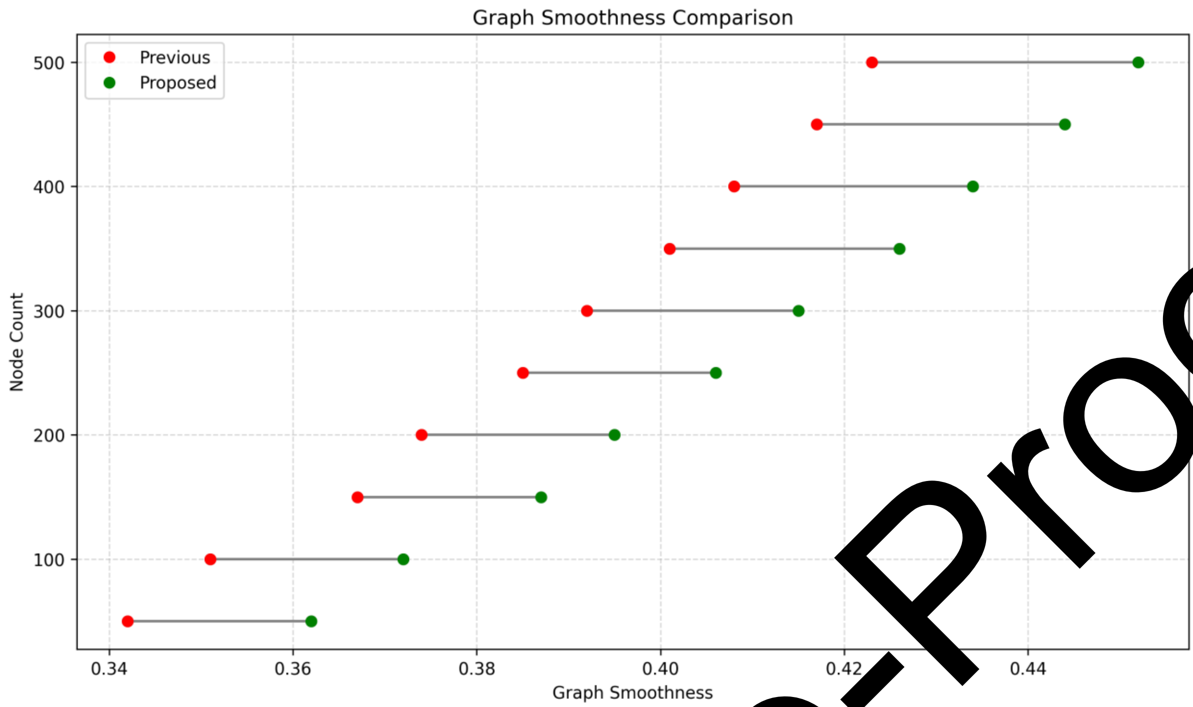


Figure 7. Graph Smoothness Score (GSS) of GNN-SCPS

The comparative analysis of the proposed GNN-SCPS model against baseline architectures, such as BPNN, CNN, and Ensemble Learning, across various node counts in vehicular SCPS, yields several critical insights. Firstly, scalability is a notable strength of GNN-SCPS; as the number of networked nodes increases, the model consistently demonstrates superior performance across all considered metrics, indicating its robustness in handling complex, large-scale vehicular environments. Secondly, the model exhibits exceptional learning robustness, with significantly higher accuracy and F1-scores, reflecting its ability to generalize well to unseen scenarios and detect behavioral anomalies in dynamic simulation environments. Thirdly, the graph-awareness embedded within the GNN-SCPS architecture, as evidenced by higher modularity and GSS values, enables effective utilization of topological relationships for optimized decision-making and intelligent routing strategies.

Collectively, these results affirm the suitability of the GNN-SCPS for real-time vehicular SCPS applications. Its seamless integration with realistic mobility and communication simulators, such as SUMO and NS-3, empowers it to model and infer patterns from heterogeneous data streams efficiently. Thus, the model not only outperforms traditional deep learning approaches in classical evaluation metrics but also excels in graph-theoretic dimensions critical to modern SCPS design. Overall, the study substantiates GNN-SCPS as a scalable, adaptive, and graph-optimized architecture poised to support the next generation of intelligent transport and cyber-physical infrastructure systems.

5. Conclusion and Future Work

The rapid evolution of Intelligent Transportation Systems (ITS) and smart cities requires scalable, adaptive, and structure-aware learning models that can handle real-time vehicular dynamics. Traditional machine learning methods often struggle to capture the complex spatiotemporal relationships inherent in Cyber-Physical Systems (CPS). In this context, graph-based DL provides a promising alternative due to its inherent capability to model non-Euclidean data structures. This study introduced GNN-SCPS, a graph neural network-integrated simulation framework that leverages NS-3 for communication modeling and SUMO for traffic mobility emulation. Through extensive evaluations against baseline models—BPNN, CNN, and Ensemble Learning—across various node densities, GNN-SCPS consistently demonstrated superior performance in terms of accuracy, F1 score, modularity, and graph smoothness. The results validate the effectiveness of GNNs in learning topological and semantic patterns for optimizing vehicular CPS.

This work sets the foundation for future deployment of GNN-driven intelligence in large-scale, real-time traffic management and smart mobility solutions.

Reference

1. Siddique, M. S., Khan, M. A. K., Ahammad, I., Nath, N., Das, J. R., & Rahman, F. (2025). An Intelligent Intrusion Detection System for Cyber-Physical Systems Using GAN-LSTM Networks. *Franklin Open*, 100281.
2. Moriano, P., Hespeler, S. C., Li, M., & Mahbub, M. (2025). Adaptive anomaly detection for identifying attacks in cyber-physical systems: A systematic literature review. *Artificial Intelligence Review*, *58*(9), 283.
3. Kocsis, I., Furján-Mansoni, B., & Balajti, I. (2025). A comprehensive review of key cyber-physical systems and assessment of their education challenges. *IEEE Access*.
4. Tariq, R., Casillas-Muñoz, F., Hassan, S., & Ramírez-Montoya, M. (2024). Synergy of Internet of Things and education: Cyber-physical systems contributing towards remote laboratories, improved learning, and school management. *Journal of Social Studies Education Research*, *15*(2), 305-352.
5. Serôdio, C., Mestre, P., Cabral, J., Gomes, M., & Branco, F. (2024). Software and architecture orchestration for process control in Industry 4.0 enabled by cyber-physical systems technologies. *Applied Sciences*, *14*(5), 2160.
6. Li, H., & Xiong, T. (2024). Cyber-Physical System Information Collection: Robot Location Method Based on QR Code. *IEEE Access*, *12*, 67046-67062.

7. Li, Q., Yang, X., Xie, X., & Liu, G. (2025). The data recovery strategy on machine learning against false data injection attacks in power cyber-physical systems. *Measurement and Control*, *58*(5), 632-642.
8. Gandhi, R. R., Inbamani, A., Divya, N., Karthik, M., & Ramya, E. (2024). Soft Computing Techniques for Cyber-Physical Systems. In *The Fusion of Artificial Intelligence and Soft Computing Techniques for Cybersecurity* (pp. 169-193). Apple Academic Press.
9. Ezechi, C., Akinsolu, M. O., Sangodoyin, A. O., Akinsolu, F. T., & Sakpere, W. (2024). Software-defined networking in cyber-physical systems. *Cyber Physical System 2.0: Communication and Computational Technologies*, 44.
10. Sagu, A., Gill, N. S., Gulia, P., Priyadarshini, I., & Chatterjee, J. M. (2024). Hybrid Optimization Algorithm for Detection of Security Attacks in IoT-Enabled Cyber-Physical Systems. *IEEE Transactions on Big Data*, *11*(1), 35-46.
11. Levytskyi, V., Kruk, P., Lopuha, O., Sereda, D., Sapaiev, V., & Matsiievskyi, O. (2024, May). Use of deep learning methodologies in combination with reinforcement techniques within autonomous mobile cyber-physical systems. In *2024 IEEE 4th International Conference on Smart Information Systems and Technologies (SIST)* (pp. 455-460). IEEE.
12. Zhang, Y., Xu, S., Wei, B., Chen, Z., Bhatti, U. A., & Huang, M. (2025). Optimizing co-simulation with the age of information in intelligent cyber-physical systems. *Internet of Things*, *31*, 101550.
13. Acharya, S., Khan, A. A., & Päivärinta, T. (2024). Interoperability levels and challenges of digital twins in cyber–physical systems. *Journal of Industrial Information Integration*, *42*, 100714.
14. Madasamy, N. S., Eldho, K. J., Senthilnathan, T., & Deny, J. (2024). A Novel Back-Propagation Neural Network for Intelligent Cyber-Physical Systems for Wireless Communications. *IETE Journal of Research*, *70*(2), 1361-1373.
15. Khan, L., Isanaka, S. P., & Liou, F. (2024). FPGA-Based Sensors for Distributed Digital Manufacturing Systems: A State-of-the-Art Review. *Sensors (Basel, Switzerland)*, *24*(23), 7709.
16. Subahi, A. F. (2024). Advancing Sustainable Cyber-Physical System Development with a Digital Twins and Language Engineering Approach: Smart Greenhouse Applications. *Technologies*, *12*(9), 147.

17. RS, P. (2024). An intelligent, dynamic cyber-physical system threat detection system for ensuring secured communication in 6G autonomous vehicle networks. *Scientific Reports*, *14*(1), 1-21.
18. Sakthi, U., Alasmari, A., Girija, S. P., Senthil, P., Qamar, S., & Hariharasitaraman, S. (2024). Smart healthcare-based cyber-physical system modeling by blockchain with cloud 6 G network and machine learning techniques. *Wireless Personal Communications*, 1-25.
19. Lo, C., Win, T. Y., Rezaeifar, Z., Khan, Z., & Legg, P. (2024, August). Digital Twins of Cyber Physical Systems in Smart Manufacturing for Threat Simulation and Detection with Deep Learning for Time Series Classification. In *2024 the 29th International Conference on Automation and Computing (ICAC)* (pp. 1-6). IEEE.
20. Banerjee, S., Jesubalan, N. G., Kulkarni, A., Agarwal, A., & Rathore, A. S. (2024). Developing cyber-physical systems and digital twin for smart manufacturing: Methodology and case study of continuous clarification. *Journal of Industrial Information Integration*, *38*, 100577.
21. Devliyal, S., Goyal, H. R., & Sharma, [illegible] (2025). iCDRET: A Dynamic Cyber Risk Estimation Technique for Intelligent Cyber-physical Systems in PCS. *The Open Bioinformatics Journal*, *18*(1).