

# Graph neural network-based prediction of cascading supply chain disruptions using real-time logistics and supplier risk data

Ramu Mannava <sup>1</sup>, Jyothinadh Nadella<sup>2</sup>, Gangadhar Reddy Bhimireddy<sup>3</sup>, Bhagam Ravi Teja<sup>4</sup>

<sup>1,2</sup> Devfi INC, USA

<sup>3</sup> Capgemini, USA

<sup>4</sup> Volantsoft Inc, USA

**Corresponding Author:** Ramu Mannava, **E-mail:** [mannava1@gmail.com](mailto:mannava1@gmail.com)

## ABSTRACT

Cascading supply chain disruptions arise when a localized failure propagates through supplier, production, transportation, and distribution dependencies, so that the eventual operational impact is determined by both the condition of individual firms and the structure of the network connecting them. Conventional risk scores and stand-alone time-series models are limited in this setting because they usually treat suppliers as independent observations and therefore fail to represent higher-order exposure. This paper develops a graph neural network-based framework for predicting near-term node-level disruptions by integrating relatively slow-moving supplier risk indicators with high-frequency logistics signals. The supply chain is represented as a directed, weighted, time-indexed graph in which nodes denote firms or facilities, edges encode material or logistics dependencies, node attributes include financial, geopolitical, concentration, quality, lead-time, and resilience indicators, and dynamic observations include delay, congestion, capacity stress, demand volatility, inventory pressure, estimated-time-of-arrival variance, and exogenous shocks. An edge-weighted graph encoder learns structural exposure, while a gated recurrent unit summarizes the temporal evolution of graph states. Because no public benchmark currently combines complete multi-tier supplier relationships with synchronized real-time logistics and verified disruption labels, the empirical section uses a reproducible semi-synthetic network designed to emulate these data-generating conditions without presenting simulated observations as real company data. On a held-out temporal test period, graph-aware models achieved stronger discrimination than non-graph baselines. The temporal GCN-GRU obtained an AUROC of 0.844, AUPRC of 0.546, recall of 0.710, and F1 score of 0.592, while a snapshot GCN achieved the highest AUROC of 0.845 and AUPRC of 0.552. Ablation analysis showed that removing network propagation reduced AUROC to 0.810 and removing real-time logistics signals reduced AUROC to 0.825. The results support a central conclusion: supply chain disruption prediction benefits materially from representing inter-firm dependency as a graph, but temporal complexity should be added only when the data resolution and propagation lags justify it. The paper concludes with deployment guidance covering graph construction, calibration, explanation, data governance, concept drift, and integration with digital supply chain twins. Graph Neural Networks, Supply Chain Disruption, Cascading Failure, Ripple Effect, Supplier Risk, Real-Time Logistics, Temporal Graph Learning, Supply Chain Resilience, Risk Propagation, Digital Supply Chain Twin.

## KEYWORDS

Graph Neural Networks, Supply Chain Disruption, Cascading Failure, Ripple Effect, Supplier Risk, Real-Time Logistics, Temporal Graph Learning, Supply Chain Resilience, Risk Propagation, Digital Supply Chain Twin.

## ARTICLE INFORMATION

**ACCEPTED:** 02 March 2025

**PUBLISHED:** 26 April 2025

**DOI:** 10.32996/fcsai.2025.4.1.8

## 1. Introduction

Global supply chains are not linear sequences of independent firms. They are interdependent production and logistics networks in which a disturbance at one supplier, port, carrier, plant, or region can change the operating conditions of many other entities. The practical consequence is the ripple effect: disruption can move through several tiers, amplify through capacity constraints and inventory shortages, and appear at downstream firms after a delay. This behavior has

been documented in the supply chain resilience literature for more than a decade and is now central to discussions of network viability, stress testing, and digital supply chain twins [1], [4]-[6]. The prediction problem is therefore fundamentally relational. A supplier that appears healthy in isolation may still face high disruption probability because its critical upstream partners are distressed, its inbound lane is congested, or its substitutes are simultaneously constrained.

Traditional supply chain risk systems commonly combine supplier scorecards, threshold rules, statistical forecasting, or supervised machine learning. These methods remain useful because they are easy to audit and can perform well when the unit of analysis is a single supplier or shipment. However, they usually flatten the network into tabular rows. A model may know that supplier A has deteriorating financial indicators and that a shipment is late, but it may not know that supplier A serves three high-centrality manufacturers, shares a logistics hub with another distressed supplier, or depends on a sub-tier source that has already failed. Bayesian networks and complex-network methods addressed part of this limitation by explicitly representing dependence and risk propagation [2], [3]. Their main challenge is that hand-specified conditional structures and parametric assumptions can become difficult to maintain as networks, operational states, and data streams evolve.

At the same time, digital logistics has expanded the set of observable risk signals. Transportation management systems, GPS and telematics feeds, automatic identification system data for maritime movements, RFID events, warehouse scans, carrier milestones, port waiting times, estimated arrival updates, and Internet of Things sensors can provide near-real-time evidence of changing operational conditions. A systematic review of IoT-based supply chain management identifies tracking, tracing, monitoring, and asset visibility as core application areas [7]. Digital supply chain twin research similarly argues that useful disruption management requires a continuously updated representation of network state rather than a static map [6]. The analytical problem has shifted from a scarcity of observations to the difficulty of integrating heterogeneous observations at the correct entities, relationships, and time scales.

Graph neural networks are a natural candidate for this problem because they learn representations by combining the attributes of each node with information received from its neighbors. Graph convolutional networks, GraphSAGE, and graph attention networks established scalable methods for learning on relational data [8]-[10]. Subsequent work generalized this idea to spatial-temporal and dynamic graphs, where both node features and relationships evolve [13]-[15]. In a supply chain context, message passing can be interpreted as learned aggregation of upstream or downstream exposure. A supplier's hidden representation can incorporate its own financial and operational condition together with the conditions of firms or logistics nodes connected to it. Multiple graph layers allow information to travel across more than one tier, which is precisely the mechanism that non-relational models omit.

By 2023, two relevant research streams had already converged conceptually even though they were not yet fully integrated in supply chain disruption prediction. The supply chain risk literature had established that disturbances propagate through interdependent networks and that resilience depends on topology, redundancy, and recovery capability [1]-[6]. In parallel, the graph learning literature had developed graph convolution, inductive neighborhood aggregation, attention, dynamic graph learning, and graph explanation methods for relational prediction [8]-[16]. Together, these foundations support the use of graph-based learning for cascading supply chain risk, while also revealing a gap: direct empirical studies that fuse multi-tier supplier dependencies with synchronized logistics telemetry remained limited within the literature available through 2023.

Despite this progress, four gaps remain important for a publishable and operational disruption prediction design. First, many supply chain risk studies use either static supplier attributes or logistics time series, rather than fusing both at the graph level. Second, some graph studies treat the network as static even though route conditions, sourcing intensity, capacity, and available alternatives change over time. Third, performance is often reported without sufficient attention to temporal leakage, class imbalance, calibration, or threshold selection, all of which matter in early-warning systems. Fourth, explainability is often treated as an optional visualization, even though procurement and risk teams need to know which upstream entities and signals caused a warning before they can act.

This paper addresses these gaps through a network-aware prediction framework that combines supplier risk data, real-time logistics telemetry, dynamic edge exposure, and temporal graph learning. The model is designed for a rolling operational setting in which observations available up to time  $t$  are used to estimate the probability that node  $i$  will be disrupted in a future horizon. The architecture uses edge-weighted graph message passing to encode cross-firm exposure and a gated recurrent unit to summarize the evolution of each node's graph representation. The same framework can be simplified to a snapshot GCN when data are low-frequency or disruption dynamics are predominantly contemporaneous. This distinction is important because additional temporal complexity should demonstrate incremental value rather than being assumed to be beneficial.

The study is organized around three research questions. RQ1 asks whether explicit supply chain topology improves near-term disruption prediction relative to models that use only node-level histories. RQ2 asks how much predictive information is contributed by real-time logistics signals, supplier risk attributes, and network propagation. RQ3 asks how model outputs can be converted into operationally defensible early-warning signals without obscuring uncertainty or causal limitations. To answer these questions without fabricating access to proprietary multi-tier supply chain data, the empirical evaluation uses a reproducible semi-synthetic benchmark with controlled cascading dynamics and a strictly chronological train-validation-test split.

The paper makes four contributions. First, it specifies a unified data model that aligns supplier risk indicators, logistics events, and inter-firm dependencies in a time-indexed directed graph. Second, it presents an edge-weighted temporal GNN for node-level disruption probability and network risk scoring. Third, it reports comparative and ablation results under a transparent semi-synthetic design, including the case where a simpler snapshot GCN slightly outperforms the temporal model on ranking metrics. Fourth, it translates model design into deployment requirements for data quality, explanation, calibration, drift monitoring, graph maintenance, and digital twin integration. The objective is not to claim a universal forecasting advantage from one architecture, but to establish a technically defensible pathway for predicting cascading disruption in real operational networks.

## **2. Background and Related Literature**

### ***2.1. Cascading disruption, the ripple effect, and network resilience***

A cascading disruption differs from an isolated operational failure because its impact is transmitted through dependence. Ivanov, Sokolov, and Dolgui describe the ripple effect as disruption propagation through a supply chain, emphasizing the trade-off among efficiency, flexibility, and resilience [1]. A lean network may perform efficiently under normal conditions yet exhibit greater vulnerability when buffers, alternative suppliers, or redundant routes are limited. This observation is important for prediction because risk is conditional on both the state of a node and the ability of the surrounding network to absorb or reroute the disturbance.

Early quantitative propagation models frequently relied on Bayesian networks. Garvey, Carnovale, and Yenyurt developed a Bayesian framework that represents interdependent supply risks and derives network-oriented risk measures [2]. Ojha et al. extended Bayesian modeling to multi-echelon risk propagation and evaluated ripple effects using fragility, service level, inventory cost, and lost sales [3]. These studies demonstrate two principles that remain relevant to GNN-based systems. First, the topology of dependence changes the distribution of outcomes. Second, propagation must be evaluated at network level rather than inferred from isolated supplier scores.

Resilience research broadens the objective beyond predicting failure to understanding resistance, adaptation, and recovery. Hosseini, Ivanov, and Dolgui classify quantitative resilience methods and emphasize the need to connect disruption modeling with the capacities that allow a network to withstand and recover [4]. Ivanov and Dolgui subsequently introduced the idea of intertwined supply network viability, which treats long-term survivability as a system property across interconnected supply chains [5]. This systems perspective discourages narrow optimization of a single firm's immediate risk score when the intervention could transfer vulnerability to another part of the network.

Earlier resilience and ripple-effect research already treated disruption propagation and recovery as processes that evolve over time rather than as single static failures [1], [4]-[6]. Network viability depends on how dependencies, capacity constraints, substitution options, and recovery actions change while a disruption unfolds. This view is conceptually aligned with dynamic graph learning because the analytical graph should not be assumed to remain unchanged during a cascading event. A predictive model must therefore distinguish relatively stable structural dependencies from dynamic edge conditions such as current route capacity, observed shipment frequency, and viable substitution.

### ***2.2. Real-time logistics data and supplier risk signals***

Supplier risk data and logistics telemetry describe different dimensions of the same operational exposure. Supplier risk data are usually slower moving and include financial health, credit deterioration, country or geopolitical exposure, quality performance, capacity concentration, cyber or compliance incidents, dependence on single sources, historical reliability, and resilience resources. These variables help estimate baseline vulnerability. By contrast, logistics observations can change by the hour or day. They include transit delay, port or terminal congestion, missed milestones, dwell time, lane capacity, carrier reliability, ETA variance, route closures, weather alerts, fuel or cost shocks, and inventory coverage. A model that uses only the first group can miss the onset of an operational event; a model that uses only the second can overreact to transient noise without considering structural vulnerability.

The growth of IoT-enabled supply chain systems makes high-frequency operational features increasingly feasible. Taj et al. review the use of GPS, RFID, NFC, sensors, and connected devices for tracking, tracing, asset management, and process

visibility [7]. Maritime automatic identification system data provide a clear example of networked telemetry because vessel movements can be aggregated into time-varying port-to-port connections, while road, air, and warehouse systems can provide analogous event streams. These data characteristics support the broader hypothesis that real-time logistics observations should be interpreted in their relational context rather than as isolated time series.

Digital supply chain twin research provides an architectural interpretation of this fusion problem. Ivanov and Dolgui define a digital supply chain twin as a computerized model representing network states at a given time and argue that combining model-based and data-driven approaches can improve disruption management [6]. Within that architecture, the present framework can be viewed as a predictive layer: ingestion systems update node and edge states, the GNN estimates emerging disruption probabilities, and scenario or optimization modules can then evaluate mitigation actions.

Data synchronization remains a major practical difficulty. Supplier master data may be updated monthly, credit indicators quarterly, shipments daily, and GPS pings every few minutes. The correct response is not to force every source into the same raw frequency. Instead, a production feature store should define event-time windows, carry forward slowly changing attributes with explicit timestamps, aggregate high-frequency signals over decision-relevant intervals, and attach missingness or freshness indicators. This allows the model to distinguish a truly low-risk observation from an old or unavailable observation.

### **2.3. Graph neural networks for relational and temporal prediction**

A graph  $G=(V,E)$  consists of nodes  $V$  and edges  $E$ . In supply chains, nodes may represent firms, facilities, products, ports, carriers, or regions, while edges represent material supply, transportation, contractual dependence, shared infrastructure, or substitution. Graph neural networks learn node representations through repeated neighborhood aggregation. Kipf and Welling's GCN formulation popularized efficient graph convolution based on normalized adjacency [8]. GraphSAGE introduced inductive neighborhood aggregation, which is attractive for networks in which new suppliers or facilities appear after training [10]. Graph attention networks assign learned importance to neighbors, allowing the model to differentiate high-exposure and low-exposure relationships [9].

Dynamic supply chains require an additional temporal mechanism. Spatio-temporal GCNs combine graph structure with time-series processing and have been effective in traffic forecasting, a domain that shares the need to model flows over connected infrastructure [15]. EvolveGCN adapts GCN parameters over time and is intended for evolving graphs [14]. Temporal Graph Networks represent dynamic graphs as sequences of time-stamped events and maintain node memories that can be updated when interactions occur [13]. These approaches suggest several viable architectures for supply chain risk, ranging from snapshot message passing at regular intervals to event-driven temporal graphs.

The architecture selected in this paper is intentionally moderate. It uses edge-weighted graph convolution at each time step followed by a GRU across recent graph states. This design provides a clean separation between spatial exposure and temporal persistence, is straightforward to implement in common deep learning frameworks, and avoids assuming that a more complex attention or event-driven architecture is automatically superior. The broader GNN literature provides several alternatives, including graph attention, evolving graph convolution, and temporal graph networks [9], [13], [14]. The incremental benefit of any additional component should be validated against simpler baselines under the same chronological data split.

Interpretability is especially important in relational models because a risk score can depend on several hops of neighbors. GNNExplainer showed that model-agnostic graph explanations can identify compact subgraphs and node features relevant to a prediction [16]. In operational supply chains, the most useful explanation is often not a global feature ranking but an attributed path: for example, a sub-tier supplier with rising distress, connected through a capacity-constrained tier-one supplier and a congested lane to the focal plant. Explanations should therefore combine feature contribution with structural contribution and should be tested for stability before being used in supplier intervention decisions.

### **2.4. Research gap and analytical position**

The literature supports the use of graph structure, temporal signals, and digital-twin data integration, but it also reveals a fragmentation problem. Risk propagation models often emphasize topology without high-frequency telemetry. Logistics forecasting models often emphasize time series without multi-tier supplier dependence. GNN studies can capture both, but some are evaluated on specialized data that are difficult to reproduce or do not explicitly separate structural supplier risk from operational logistics stress. The core analytical position of this paper is that cascading disruption prediction should be treated as multimodal dynamic graph classification with explicit attention to temporal evaluation and uncertainty.

This position also requires restraint. A GNN can learn predictive association along network edges, but it does not automatically identify causal transmission. Supplier relationships can correlate with unobserved geographic, sectoral, or

ownership factors. Likewise, learned edge importance is not equivalent to a causal intervention effect. The framework therefore uses the language of predicted exposure and propagation pathways, reserving causal claims for settings with appropriate experimental, quasi-experimental, or structural identification. This distinction is important for publication quality and for responsible managerial use.

Table 1. Data domains for graph-based supply chain disruption prediction.

| Data domain          | Representative variables                                                                                      | Graph role                                         | Typical update rate              |
|----------------------|---------------------------------------------------------------------------------------------------------------|----------------------------------------------------|----------------------------------|
| Supplier risk        | Financial distress, country risk, quality failures, concentration, lead-time variability, resilience capacity | Primarily node features and baseline vulnerability | Weekly to quarterly              |
| Real-time logistics  | Shipment delay, dwell time, congestion, capacity deficit, ETA variance, carrier reliability                   | Dynamic node and edge features                     | Minutes to daily                 |
| Network structure    | Supplier-customer links, bill-of-material dependence, volume share, alternative sources                       | Directed edges and structural weights              | Daily to monthly or event-driven |
| External events      | Weather, geopolitical alerts, strikes, sanctions, natural hazards, cyber incidents                            | Node, edge, region, or global context              | Event-driven                     |
| Operational outcomes | Disruption onset, late delivery, stock-out, line stop, recovery time                                          | Training labels and severity targets               | Event-driven                     |

### 3. Problem Formulation

#### 3.1. Dynamic supply chain graph

Let the supply network at decision time  $t$  be represented by a directed weighted graph  $G_t=(V_t, E_t, X_t, W_t)$ .  $V_t$  is the set of active entities,  $E_t$  is the set of directed dependency edges,  $X_t$  contains node features, and  $W_t$  contains time-dependent edge weights. A directed edge  $j$  to  $i$  indicates that node  $i$  depends on node  $j$  for material, capacity, transportation, or another modeled service. Direction therefore follows the assumed path of disruption transmission. Reverse edges can be added separately when downstream feedback, financial contagion, or information effects are modeled.

The node feature vector  $x_{i,t}$  is partitioned into relatively static supplier risk attributes  $s_{i,t}$  and dynamic logistics attributes  $l_{i,t}$ . Static does not mean immutable; it means that the variable changes more slowly than operational telemetry. The resulting feature vector can be written as  $x_{i,t}=[s_{i,t} \parallel l_{i,t} \parallel q_{i,t}]$ , where  $q_{i,t}$  contains data-quality indicators such as observation age, missingness, or source reliability. Explicit freshness features are useful because stale supplier ratings and absent telemetry should not be silently treated as current normal conditions.

Each edge weight  $w_{ji,t}$  represents the effective exposure of node  $i$  to node  $j$  at time  $t$ . A production implementation can construct this weight from purchase share, criticality, substitutability, lane reliability, and current capacity stress. The exact formula should be domain-specific. A useful starting point is a nonnegative composite score that increases with dependency and current logistics stress and decreases with available substitution. Because supplier shares and route utilization may be confidential or partially observed, the system should also retain an uncertainty estimate for edge weights.

$$G_t = (V_t, E_t, X_t, W_t), x_{i,t} = [s_{i,t} \parallel l_{i,t} \parallel q_{i,t}] \quad (1)$$

$$w_{ji,t} = \text{Dependency}_{ji} \times \text{Exposure}_{ji,t} \times (1 - \text{Substitutability}_{ji,t}) \quad (2)$$

#### 3.2. Prediction targets and cascade risk

The primary task is binary node-level disruption prediction. Given a lookback window of  $L$  graph states, the model estimates  $p_{i,t+h}=P(y_{i,t+h}=1 \mid G_{t-L+1}, \dots, G_t)$ , where  $h$  is the prediction horizon. The disruption label should be defined operationally before model development. Examples include a missed service-level threshold, confirmed supplier shutdown, production line stop, stock-out caused by inbound failure, or a shipment delay exceeding a business-defined tolerance. Mixing several meanings of disruption in one label can make performance difficult to interpret.

A secondary task is network risk scoring. Node probabilities can be aggregated using business criticality  $c_i$  to produce a portfolio-level risk index  $R_t$ . This score is not intended to replace node warnings. Its purpose is to summarize expected

concentration of exposure and to support stress monitoring across plants, product families, or regions. A network-level alarm can be triggered when predicted risk is both elevated and structurally concentrated in critical nodes, because the same average probability can imply different operational consequences depending on where risk is located.

A third output is a candidate cascade explanation. For a high-risk target node, the system should identify influential upstream nodes, edges, and features. This can be derived from learned attention if the model uses attention, from gradient-based or perturbation attribution, or from a model-agnostic graph explainer [16]. The explanation is a diagnostic aid rather than proof of causation. It should be presented with timestamps and data freshness so that an analyst can verify whether the indicated supplier or logistics condition is still current.

$$p_{i,t+h} = P(y_{i,t+h} = 1 \mid G_{t-L+1}, \dots, G_t) \quad (3)$$

$$R_t = (\sum_i c_i p_{i,t}) / (\sum_i c_i) \quad (4)$$

### 3.3. Data alignment and leakage control

Temporal leakage is a serious risk in disruption prediction. Features must be generated using only observations that would have been available at the prediction timestamp. For example, a final shipment arrival time cannot be used to predict a delay that was not yet known, and a supplier rating revised after an incident cannot be backfilled into earlier records. Event-time feature engineering should therefore use point-in-time joins and should preserve the original publication time of external risk data.

The network itself can also leak future information. If a supplier relationship is added to the graph based on transactions that occurred after the prediction period, the model can benefit from future topology. Chronological graph snapshots should be reconstructed from effective dates, and discontinued or newly created relationships should enter the graph only when they become observable. Similar care is needed when building labels for cascading events, because downstream consequences occurring after the horizon should not be used to define upstream features.

A publishable evaluation should split data by time, not randomly by node-time rows. Random splitting can place nearly identical temporal neighborhoods in training and test sets and substantially overstate generalization. The evaluation in Section 5 therefore uses a chronological training period, a later validation period for threshold and hyperparameter selection, and the most recent period as the final test set.

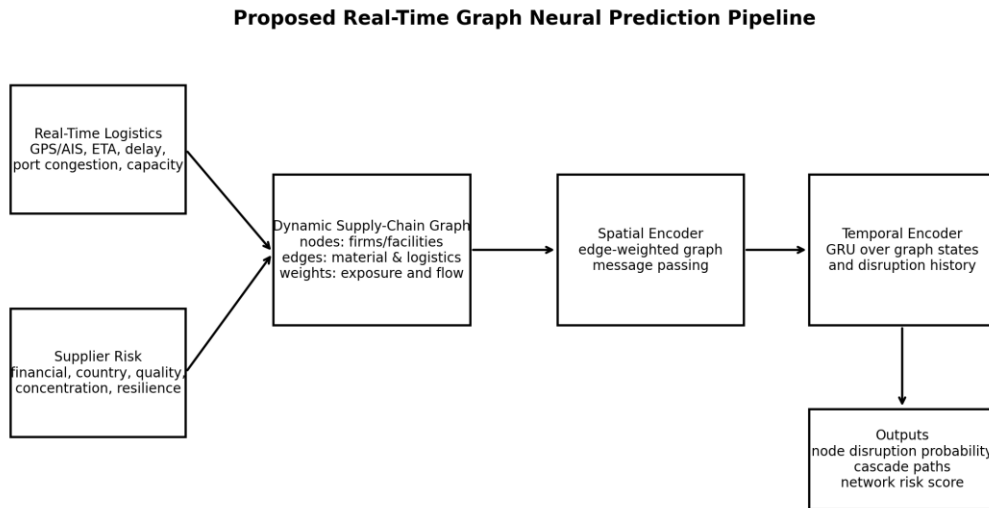


Figure 1. Proposed data-to-prediction pipeline for real-time graph neural disruption forecasting.

## 4. Proposed Graph Neural Network Method

### 4.1. Edge-weighted spatial encoder

At each time step, the spatial encoder updates a node representation by combining a transformed version of the node's own features with a weighted aggregate of incoming neighbors. Let  $h_{i,t}^{(0)} = x_{i,t}$ . For layer  $l$ , the incoming message  $m_{i,t}^{(l)}$  is the weighted sum of transformed neighbor representations. The edge weights are normalized over the incoming neighborhood so that the model remains numerically stable when nodes have different numbers of suppliers. A separate self transformation preserves entity-specific information that may not be explained by neighbors.

This design is intentionally edge-aware. In a supply chain, two suppliers connected to the same manufacturer should not automatically contribute equal exposure. A supplier representing 60 percent of a critical component or using the only

viable lane should carry more structural weight than a supplier with a small spend share and several substitutes. Dynamic logistics factors can modify the edge weight so that a relationship becomes more influential when the associated route is congested or capacity constrained.

Two graph layers are used in the proof-of-concept experiment. The first aggregates direct supplier exposure; the second allows indirect information to travel approximately two hops, subject to the direction and density of the graph. More layers are not automatically better. Deep message passing can over-smooth node embeddings and can mix distant relationships that are operationally weak. The receptive field should correspond to a plausible number of tiers for the prediction horizon.

$$m_{i,t}^{(l)} = \sum_{j \in N_i(t)} a_{ji,t} W_n^{(l)} h_{j,t}^{(l)} \quad (5)$$

$$h_{i,t}^{(l+1)} = \text{ReLU}(W_s^{(l)} h_{i,t}^{(l)} + m_{i,t}^{(l)}) \quad (6)$$

#### 4.2. Temporal encoder

The spatial encoder produces a graph-conditioned representation  $h_{i,t}$  for every node and time step. A gated recurrent unit then processes the sequence  $h_{i,t-L+1}, \dots, h_{i,t}$ . GRUs are selected because they provide temporal memory with fewer parameters than a conventional LSTM and are sufficient for a proof-of-concept with a short lookback window. The hidden state  $z_{i,t}$  summarizes persistent or delayed patterns such as rising congestion, repeated late milestones, or upstream disruptions that have not yet reached the focal node.

The output layer maps  $z_{i,t}$  to a disruption probability through a sigmoid function. For operational use, raw neural probabilities should be evaluated for calibration. Modern neural networks can be poorly calibrated even when their ranking performance is strong, and simple post-hoc methods such as temperature scaling can improve probability reliability [17]. Calibration should be performed on a validation set that is later in time than the training set and should be rechecked after material changes in prevalence or operating conditions.

$$z_{i,t} = \text{GRU}(h_{i,t}, z_{i,t-1}) \quad (7)$$

$$p_{i,t+1} = \text{sigmoid}(w_o^T z_{i,t} + b_o) \quad (8)$$

#### 4.3. Objective function and class imbalance

Disruption events are usually less common than normal operations, so an unweighted loss can favor the majority class. The model is trained with weighted binary cross-entropy, where positive events receive a larger weight derived from the class ratio in the training period. This changes the training objective but does not determine the final alert threshold. Thresholds are selected separately on validation data according to the decision objective, for example maximum F1, minimum expected cost, or a required recall level for critical suppliers.

Evaluation includes both AUROC and area under the precision-recall curve. AUROC measures ranking across thresholds, while AUPRC is especially informative when positives are rare because it emphasizes precision among detected positives [18]. Precision, recall, F1, accuracy, and Brier score are also reported at a validation-selected threshold. Brier score is included because a model can rank cases well yet produce poorly calibrated probabilities.

$$L = - \sum_{i,t} [\beta y_{i,t} \log(p_{i,t}) + (1-y_{i,t}) \log(1-p_{i,t})] + \lambda ||\theta||_2^2 \quad (9)$$

#### 4.4. Explainability and path attribution

A practical warning should answer three questions: which entity is at risk, what evidence is driving the prediction, and through which dependencies might the risk propagate. Feature attribution can identify whether delay, congestion, inventory stress, financial vulnerability, or another input contributed strongly. Structural attribution can identify influential upstream nodes or edges. GNNExplainer provides one general approach for extracting a compact explanatory subgraph [16], while edge occlusion offers a simpler operational alternative: temporarily remove or down-weight a candidate edge and measure how much the predicted probability changes.

Explanations should be evaluated for fidelity and stability. If small perturbations in noncritical data produce completely different supplier paths, the explanation is not reliable enough for procurement action. Stability can be measured across nearby time windows or model retraining runs. In regulated or high-stakes environments, the system should retain an audit record containing the model version, graph snapshot, feature timestamps, probability, threshold, and explanation shown to the analyst.

#### 4.5. Deployment architecture

A production implementation can separate the system into five services. The ingestion layer collects supplier master data, risk scores, purchase orders, shipment events, tracking feeds, and external event data. The entity-resolution layer maps source-specific identifiers to canonical firms, facilities, ports, lanes, and products. The graph service maintains effective-dated nodes and edges and computes structural weights. The feature service produces point-in-time node and edge

tensors. The model service generates disruption probabilities and explanations and sends them to a risk dashboard or digital twin.

This separation matters because many apparent model errors originate in entity resolution or graph maintenance rather than neural optimization. Duplicate suppliers, incorrect parent-subsidary mapping, stale lane relationships, and missing sub-tier connections can distort propagation. Model monitoring should therefore track not only predictive metrics but also graph health metrics such as node coverage, edge freshness, percentage of spend mapped to known suppliers, connected-component changes, and the share of critical products with multi-tier visibility.

Table 2. Proposed model components and operational rationale.

| Component                   | Function                                                 | Operational rationale                                     |
|-----------------------------|----------------------------------------------------------|-----------------------------------------------------------|
| Point-in-time feature store | Aligns supplier and logistics observations to event time | Prevents future information from leaking into predictions |
| Dynamic directed graph      | Represents supplier and logistics dependence             | Preserves propagation direction and multi-tier exposure   |
| Edge-weighted GCN           | Aggregates neighbor risk using dependency strength       | Differentiates critical and weak relationships            |
| GRU temporal encoder        | Summarizes recent graph states                           | Captures delayed propagation and persistent stress        |
| Probability calibration     | Maps scores to usable risk probabilities                 | Supports thresholds and expected-loss decisions           |
| Graph explanation layer     | Identifies influential features, nodes, and paths        | Supports analyst review and mitigation planning           |

## 5. Experimental Design

### 5.1. Semi-synthetic benchmark and scope

A major limitation in this research area is the absence of a widely available dataset that simultaneously contains verified multi-tier supplier links, synchronized real-time logistics signals, supplier risk attributes, and disruption onset labels. Complete supplier networks and transaction histories are often commercially sensitive, while public logistics datasets typically cover only part of the dependency structure required for multi-tier propagation analysis. Rather than inventing a proprietary source, this paper evaluates the method on a semi-synthetic benchmark generated specifically to reproduce the statistical and structural conditions of the proposed use case. The benchmark is intended as a proof of concept and not as evidence of field performance at a named company.

The simulated network contains 96 nodes distributed across four tiers and 242 directed supply links. Each nonterminal node is connected to one or more downstream entities, with occasional skip-tier links to represent direct sourcing across tiers. Static node features represent financial distress, geopolitical exposure, supplier concentration, quality risk, lead-time variability, and resilience capacity. Dynamic features represent shipment delay, congestion, transport capacity deficit, demand volatility, inventory stress, ETA variance, weather or external shock intensity, and the node's recent disruption state.

Cascading labels are generated from a nonlinear hazard process. A node's probability of disruption depends on its baseline supplier vulnerability, its current logistics stress, global and local exogenous shocks, and the disruption state of incoming suppliers over multiple recent time steps. The process also includes persistence and stochastic recovery. This construction creates the specific challenge the model is intended to solve: a downstream disruption can be predicted more accurately when the learner sees both the target's time series and the states of connected upstream nodes.

The full simulation contains 260 time steps and 24,960 node-time observations. The disruption rate is 12.4 percent, creating meaningful class imbalance without making positive events extremely rare. A six-step lookback window is used to predict the next time step. Windows are split chronologically into 177 training windows, 38 validation windows, and 39 test windows. Feature standardization uses training data only. Alert thresholds are selected on validation data and then frozen before evaluation on the test period.

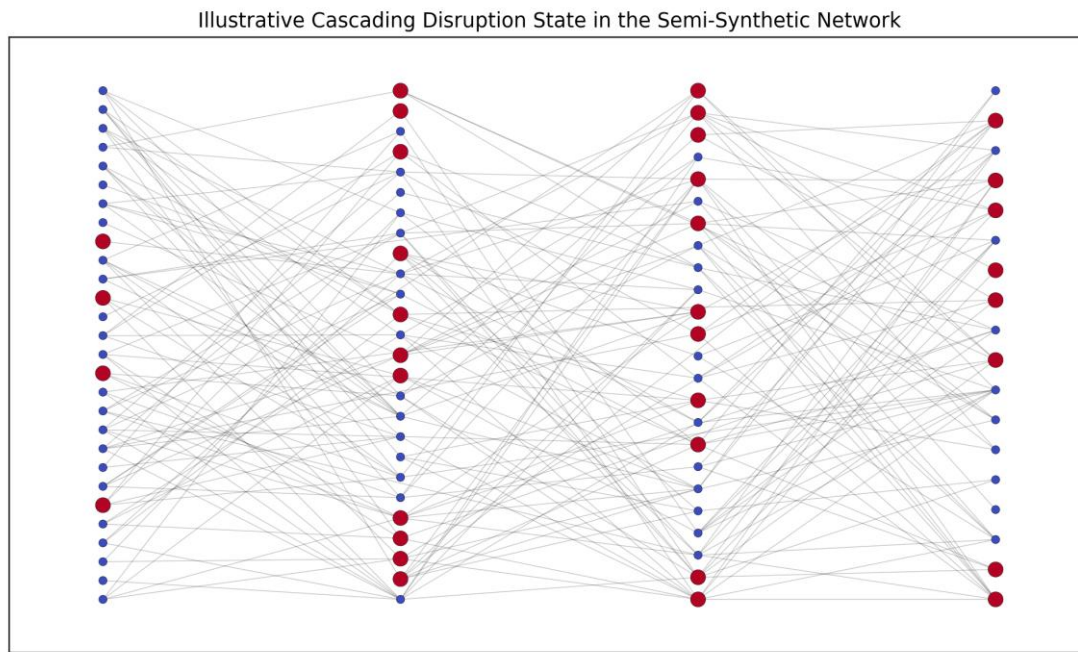


Figure 2. Illustrative disruption state in the semi-synthetic four-tier supply network. Enlarged warm-colored nodes are disrupted at the selected time step.

Table 3. Semi-synthetic benchmark characteristics.

| Characteristic                | Value                                    |
|-------------------------------|------------------------------------------|
| Supply-chain nodes            | 96                                       |
| Directed dependency links     | 242                                      |
| Time steps                    | 260                                      |
| Node-time observations        | 24,960                                   |
| Prediction lookahead          | 6 time steps                             |
| Overall disruption prevalence | 12.4%                                    |
| Training windows              | 177                                      |
| Validation windows            | 38                                       |
| Test windows                  | 39                                       |
| Split strategy                | Chronological, no random node-time split |

## 5.2. Baseline models

Five models are compared. Logistic regression uses the latest feature vector together with window means and first-to-last changes. Random forest uses the same tabular representation and provides a nonlinear non-graph baseline. A node-level GRU uses the full temporal sequence for each node but has no access to neighbor information. A snapshot GCN uses the most recent graph and node features without recurrent temporal processing. The temporal GCN-GRU is the proposed full model, combining two graph message-passing layers with a recurrent temporal encoder.

The comparison is deliberately designed to isolate sources of value. Logistic regression tests whether simple trends are sufficient. Random forest tests nonlinear tabular interactions. The node GRU tests temporal sequence learning without topology. The snapshot GCN tests topology without explicit temporal memory. The temporal GCN-GRU tests the combination. Because the objective is methodological comparison rather than exhaustive architecture search, hyperparameter tuning is modest and is performed only on the validation period.

Table 4. Baseline and proposed model configurations.

| Model               | Network information | Temporal sequence       | Primary role                        |
|---------------------|---------------------|-------------------------|-------------------------------------|
| Logistic regression | No                  | Summary statistics only | Transparent linear baseline         |
| Random forest       | No                  | Summary statistics only | Nonlinear tabular baseline          |
| Node GRU            | No                  | Yes                     | Temporal-only neural baseline       |
| Snapshot GCN        | Yes                 | No                      | Graph-only baseline                 |
| Temporal GCN-GRU    | Yes                 | Yes                     | Proposed spatiotemporal graph model |

### 5.3. Training and evaluation protocol

Neural models are optimized with Adam, L2 weight decay, dropout in the temporal GNN, and early stopping based on validation AUROC. Weighted binary cross-entropy compensates for the positive-negative class ratio. The graph adjacency is row-normalized after reversing the supplier-to-customer edge representation into an incoming-neighbor aggregation matrix and adding self-loops. This means that each receiver aggregates a convex combination of its own state and the states of supplying neighbors, with larger dependency and logistics exposure receiving greater weight.

The decision threshold for each model is selected on the validation period to maximize F1. In a real deployment, threshold selection should reflect explicit economic costs. Missing a disruption at a single-source strategic supplier may be far more costly than investigating a false positive, while a low-criticality tail supplier may justify a different threshold. The validation-optimized F1 threshold is used here only to provide a common comparison across models.

The experiment was implemented in Python using PyTorch for neural networks, scikit-learn for tabular baselines and metrics, and NetworkX for graph construction and visualization. A fixed random seed was used for reproducibility. The benchmark generation code and parameterization are suitable for inclusion as supplementary material in a journal submission.

## 6. Results

### 6.1. Comparative predictive performance

Table 5 reports performance on the held-out temporal test period. The two graph-aware models provide the strongest ranking performance. The snapshot GCN records the highest AUROC at 0.845 and AUPRC at 0.552. The temporal GCN-GRU is nearly identical in AUROC at 0.844 and reaches an AUPRC of 0.546. At the validation-selected operating threshold, the temporal model achieves the highest F1 score, 0.592, with recall of 0.710. These results indicate that structural information is consistently useful, while the additional recurrent layer provides only a modest operating-point benefit in this particular benchmark.

The result is important because it argues against complexity for its own sake. The simulated cascade process contains delayed effects, so a temporal graph model is theoretically appropriate. Nevertheless, the latest graph snapshot already contains substantial evidence through current delay, congestion, inventory stress, and previous disruption state. The snapshot GCN can therefore recover much of the relevant signal. In production, a temporal architecture should be justified by higher-frequency observations, longer propagation lags, or a meaningful improvement in business metrics.

Among non-graph models, logistic regression achieves AUROC 0.815 and AUPRC 0.532, while the node GRU achieves AUROC 0.812 and AUPRC 0.516. Random forest yields AUROC 0.796 and AUPRC 0.507. The node GRU does not outperform logistic regression, which reinforces the conclusion that temporal modeling without network context is not sufficient for a process generated partly by upstream propagation.

Table 5. Held-out temporal test performance.

| Model               | AUROC | AUPRC | Precision | Recall | F1    | Accuracy | Brier |
|---------------------|-------|-------|-----------|--------|-------|----------|-------|
| Logistic regression | 0.815 | 0.532 | 0.560     | 0.613  | 0.585 | 0.854    | 0.177 |
| Random forest       | 0.796 | 0.507 | 0.598     | 0.561  | 0.579 | 0.862    | 0.124 |
| Node GRU            | 0.812 | 0.516 | 0.554     | 0.610  | 0.581 | 0.851    | 0.183 |
| Snapshot GCN        | 0.845 | 0.552 | 0.484     | 0.739  | 0.585 | 0.823    | 0.165 |
| Temporal GCN-GRU    | 0.844 | 0.546 | 0.507     | 0.710  | 0.592 | 0.835    | 0.169 |

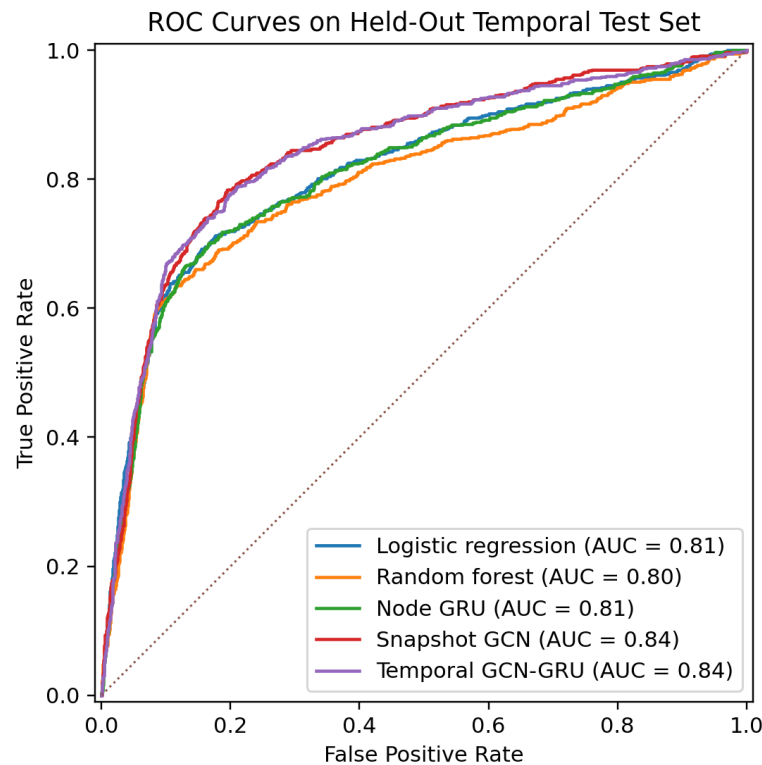


Figure 3. ROC curves for graph and non-graph models on the chronological test period.

### 6.2. Ablation analysis

Ablation analysis evaluates the marginal importance of the three information groups central to the research question. Removing supplier-risk features from the temporal GCN-GRU lowers AUROC from 0.844 to 0.839 and AUPRC from 0.546 to 0.531. Removing real-time logistics variables produces a larger decrease, to AUROC 0.825 and AUPRC 0.509. Replacing the supply network with identity-only self connections, which effectively removes propagation, reduces AUROC to 0.810. The ordering is consistent with the benchmark design: dynamic logistics and inter-firm exposure contain the most immediate signal, while supplier risk provides baseline vulnerability that improves ranking more modestly.

Threshold-based F1 does not decrease monotonically in every ablation because the threshold was selected on a relatively small validation period and because removing features can change score calibration. For this reason, AUROC and AUPRC are treated as the primary ablation statistics. This is also a useful methodological lesson for applied supply chain research: a single F1 value can hide changes in ranking and probability quality, especially when thresholds are reoptimized after each model change.

Table 6. Ablation results for the temporal GCN-GRU.

| Variant                         | AUROC | AUPRC | F1    | Recall |
|---------------------------------|-------|-------|-------|--------|
| Full model                      | 0.844 | 0.546 | 0.592 | 0.710  |
| No supplier-risk features       | 0.839 | 0.531 | 0.600 | 0.716  |
| No real-time logistics features | 0.825 | 0.509 | 0.559 | 0.525  |
| No network propagation          | 0.810 | 0.522 | 0.587 | 0.604  |

### 6.3. Interpretation of the model comparison

The clearest empirical pattern is the value of topology. Both graph models outperform the node GRU in AUROC by more than three percentage points. The gain is not enormous, but it is systematic and occurs under a chronological test split. In operational terms, this means that an analyst who observes the same local supplier and logistics history can obtain a better risk ranking when the model also knows which upstream nodes are disrupted and how strongly the focal node depends on them.

The second pattern is the importance of dynamic logistics signals. The no-logistics ablation loses roughly 0.019 AUROC, a larger reduction than the no-supplier-risk ablation. This should not be interpreted as evidence that financial or strategic

supplier risk is unimportant. The prediction horizon is one time step, so immediate operational variables naturally dominate near-term onset. For longer horizons, baseline supplier vulnerability and external macro risk may become more influential. A production system should therefore match the feature mix to the decision horizon.

The third pattern concerns calibration. Random forest has a lower Brier score than the neural models despite weaker ranking performance. This shows why operational deployment should not equate the best AUROC with the best probability estimates. If disruption probabilities are used in expected-loss calculations, model outputs should be recalibrated and tested by supplier criticality segment. Ranking and calibration are separate properties.

#### **6.4. Cascade-level scenario interpretation**

Figure 2 illustrates a high-disruption state from the semi-synthetic test period. Failures occur across several tiers rather than remaining concentrated at a single origin. This is the type of pattern that graph message passing is designed to represent. A downstream node can receive messages from several stressed suppliers at the same time, and a second graph layer can incorporate indirect stress from sub-tier entities. The model therefore estimates conditional exposure from the local graph neighborhood instead of relying only on a global shock variable.

In a real system, a high-risk prediction would be accompanied by a ranked explanatory subgraph. Suppose a manufacturer receives a probability above its action threshold. The explanation might show that two upstream suppliers have elevated disruption scores, that one supplies a high-share component, and that the common inbound lane has deteriorating ETA variance and congestion. The risk team can then distinguish interventions such as expediting, inventory reallocation, alternate sourcing, carrier substitution, or supplier engagement. This workflow is more actionable than a black-box alert with no propagation context.

### **7. Discussion**

#### **7.1. Answering the research questions**

RQ1 asked whether explicit topology improves near-term disruption prediction. Within the proof-of-concept benchmark, the answer is yes. Graph-aware models achieved higher AUROC and AUPRC than logistic regression, random forest, and a temporal node GRU. The ablation that removes network propagation also produces the largest AUROC decline among the information-removal tests. These results are consistent with prior Bayesian supply chain propagation research [2], [3] and with the broader graph neural network principle that neighborhood structure can carry predictive information unavailable in isolated node histories [8]–[12].

RQ2 asked about the incremental contribution of real-time logistics and supplier risk. Both contribute, but their influence differs at the one-step horizon. Logistics features provide the larger ranking gain because they encode immediate operational deterioration. Supplier risk features improve AUROC and AUPRC more modestly but provide structural vulnerability that can be important when operational evidence is incomplete or when the prediction horizon is extended. The result supports a layered risk model in which supplier risk establishes prior vulnerability and logistics signals update near-term hazard.

RQ3 asked how outputs can be made operationally defensible. The key requirements are point-in-time data, explicit graph definitions, calibrated probabilities, segment-specific thresholds, and explanations that identify influential nodes, edges, and features. A production warning should not be interpreted as certainty that a specific upstream partner caused the disruption. It should be interpreted as evidence that the observed network state is associated with elevated risk and that the indicated dependency path deserves investigation.

#### **7.2. Managerial and operational implications**

The model can support at least four decision levels. At the supplier level, procurement teams can prioritize outreach, qualification, and contingency actions for high-risk suppliers. At the shipment level, logistics teams can focus expediting or rerouting on flows whose delay is likely to propagate into critical production nodes. At the product or plant level, network risk scores can identify concentrated exposure across several suppliers. At the executive level, scenario analysis can estimate how a regional or infrastructure shock may spread through the portfolio.

The biggest operational benefit is prioritization rather than automation. Large enterprises can generate thousands of risk signals each day, many of which are weak or redundant. A graph model can rank alerts by combining local severity with downstream importance. A moderate-risk supplier that feeds a highly central plant may warrant more attention than a high-risk supplier serving a noncritical product with many alternatives. Criticality weights and expected loss should therefore be kept distinct from predicted probability.

A practical policy can define escalation bands rather than a single universal threshold. Low-risk scores remain in monitoring. Medium scores trigger evidence review. High scores trigger supplier or carrier contact and scenario

simulation. Critical scores at strategically important nodes trigger pre-approved mitigation actions. Such a design reduces alert fatigue and keeps humans responsible for interventions that involve commercial, legal, or safety consequences.

Table 7. Example decision policy for converting model output into supply chain actions.

| Risk band | Illustrative condition                                       | Primary action                                                               | Required evidence                                          |
|-----------|--------------------------------------------------------------|------------------------------------------------------------------------------|------------------------------------------------------------|
| Monitor   | Low calibrated probability and low criticality               | Continue automated monitoring                                                | Fresh data and no material cascade path                    |
| Watch     | Moderate probability or worsening trend                      | Analyst review and supplier contact if needed                                | Recent logistics anomaly or supplier-risk deterioration    |
| Warning   | High probability or concentrated upstream exposure           | Evaluate expedite, alternate lane, inventory transfer, or alternate supplier | Verified graph path, current capacity and inventory status |
| Critical  | High probability at a single-source or high-criticality node | Activate contingency plan and executive escalation                           | Cross-source confirmation, scenario impact, action owner   |

### 7.3. Data governance, security, and model risk

Supplier and logistics graphs contain commercially sensitive information. Multi-tier maps can reveal strategic sourcing, volumes, capacities, and concentration. Access should therefore follow least-privilege principles, and graph exports should be controlled as carefully as procurement contracts. When external supplier-risk services are integrated, licensing restrictions and permitted uses must be reviewed. Data lineage should record the source and timestamp of each feature so that alerts can be reproduced during audit or incident review.

Entity resolution is another governance risk. The same company can appear under different legal names, subsidiaries, sites, or vendor identifiers. Incorrect merges can create false edges; missed merges can hide concentration. A production graph should distinguish legal entity, facility, and corporate parent where these roles matter. Confidence scores for entity matches can be retained as model inputs or used to suppress low-confidence propagation paths.

Model risk controls should include out-of-time validation, calibration checks, subgroup performance, drift monitoring, and challenger models. A simple logistic or tree model remains useful as a sanity-check benchmark even after a GNN is deployed. If the graph model loses its advantage, the organization should investigate whether network data became stale, feature distributions shifted, or the operating regime changed. Complex models should earn their maintenance cost continuously.

### 7.4. Relation to digital supply chain twins

The proposed GNN is best interpreted as one analytical component of a digital supply chain twin rather than a complete twin by itself. Digital twin research emphasizes a continuously synchronized representation of the physical network, real-time visibility, disruption analysis, and decision support [6]. The GNN provides the predictive state estimator: it converts incoming telemetry and supplier information into risk probabilities and propagation explanations. A simulation or optimization layer can then test actions such as rerouting, reallocating inventory, changing order quantities, or activating backup suppliers.

This closed-loop design is also a natural place to collect intervention outcomes. If an alert leads to an expedite or supplier switch, the system should record the action and resulting performance. Over time, this creates data for evaluating not just predictive accuracy but decision effectiveness. The most valuable future system may therefore combine graph prediction with causal or counterfactual models that estimate how candidate interventions change expected loss.

### 7.5. Limitations

The first limitation is the semi-synthetic evaluation. The benchmark is transparent and useful for controlled comparison, but it cannot reproduce the full heterogeneity, missingness, strategic behavior, and reporting delay of real supplier networks. The reported metrics should therefore be read as proof-of-concept evidence, not as expected field accuracy. External validation on a real multi-tier network is required before publication claims about operational effectiveness can be generalized.

The second limitation is graph completeness. Many organizations have strong visibility into tier-one suppliers but incomplete visibility into sub-tier dependencies. A GNN cannot propagate information along edges that are unknown. Probabilistic graph completion or supplier-discovery methods may help, but inferred edges introduce their own uncertainty. The system should distinguish verified contractual relationships from inferred or indirect relationships.

The third limitation is causal ambiguity. Message passing learns predictive dependence and can highlight plausible propagation paths, but the learned path may reflect common exposure to a region, market, or infrastructure asset rather

than direct transmission. Decisions that require causal attribution should use complementary evidence. The fourth limitation is temporal granularity. A six-step GRU is appropriate for the simulated process but may be too short for strategic supplier distress or too long for minute-level logistics anomalies. Production horizons should be selected from business lead times and intervention latency.

Finally, the current formulation predicts binary onset and does not jointly model disruption severity, duration, recovery time, or competing disruption types. Supply chain resilience research treats disruption impact, recovery, survivability, and network viability as connected dimensions of risk management [1], [4], [5]. Extending the graph model to multi-task prediction could therefore improve the information available for action by estimating not only whether a disruption is likely, but also its expected severity and recovery characteristics.

## 8. Future Research Directions

The first priority is real-world multi-company validation. A strong next study would combine purchase-order or bill-of-material relationships, supplier financial and risk indicators, shipment milestone data, and verified disruption outcomes across several years. Evaluation should preserve chronology and, where possible, include an external network or later time period. Equivalent public benchmarks for multi-tier firm networks remain scarce, so reproducible evaluation protocols and privacy-preserving data partnerships are important research priorities.

Second, future models should represent heterogeneous node and edge types explicitly. Suppliers, plants, ports, warehouses, carriers, products, and regions have different state variables and different propagation mechanisms. Relational GNNs or heterogeneous graph transformers can learn type-specific transformations rather than forcing all dependencies into a single adjacency matrix. Product-level bill-of-material relationships are especially important because a low-spend component can be operationally critical if it has no substitute.

Third, uncertainty should be modeled directly. Supplier network edges can be incomplete, risk ratings can be stale, and logistics signals can be noisy. Bayesian graph models, ensembles, conformal prediction, or probabilistic latent graphs can provide prediction intervals or uncertainty sets. In this analytical framework, structural and feature uncertainty are treated as central rather than peripheral. A risk dashboard should distinguish a high probability supported by strong evidence from a high probability driven by sparse or uncertain data.

Fourth, explainability should move from generic feature attribution toward counterfactual decision support. The useful question for managers is not only why the model predicts disruption, but what feasible action is expected to reduce risk. Counterfactual graph analysis could estimate how predicted risk changes if a lane is restored, an alternate supplier is activated, or inventory coverage increases. Graph explanation methods can identify influential subgraphs and features [16], while digital supply chain twin concepts provide a natural environment in which candidate interventions can be simulated before operational execution [6].

Fifth, temporal graph architectures should be matched to event frequency. Regular snapshot models are appropriate when supply relationships and logistics indicators are updated on fixed intervals. Event-driven Temporal Graph Networks may be preferable when each purchase order, shipment milestone, or supplier event arrives asynchronously [13]. EvolveGCN offers another approach when node sets or graph structure change substantially over time [14]. Comparative work should evaluate not just predictive accuracy but latency, memory cost, ease of retraining, and robustness to missing events.

Sixth, research should connect prediction with resilience optimization. A warning system is valuable only if interventions reduce service loss or recovery time. A digital twin can use GNN probabilities as state estimates and then simulate mitigation portfolios under capacity, cost, and contractual constraints. The resulting objective can optimize expected service loss, recovery time, cost, carbon impact, or a multi-criteria combination. This would close the loop between risk sensing and resilience action envisioned in digital twin research [6].

Finally, evaluation standards should become more rigorous. Future studies should report point-in-time feature construction, chronological splits, prevalence, calibration, decision thresholds, and baselines that isolate the value of topology. When data are confidential, authors can still publish graph schemas, feature definitions, synthetic generators, and evaluation code. Reproducibility is particularly important in a field where proprietary data otherwise make strong performance claims difficult to verify independently.

## 9. Conclusion

Cascading supply chain disruption prediction is a relational forecasting problem. The risk faced by a supplier, plant, port, or distribution node depends not only on its own condition but also on the condition of connected entities and the ability of the network to absorb or reroute stress. This paper develops a graph neural network framework that fuses supplier risk

indicators, real-time logistics features, and directed dependency structure to estimate near-term disruption probability and network exposure.

The semi-synthetic proof-of-concept provides controlled evidence for the value of this representation. Graph-aware models outperform non-graph baselines on ranking metrics, and removing network propagation produces the largest reduction in AUROC. Real-time logistics features also contribute materially, while supplier risk attributes add baseline vulnerability information. The temporal GCN-GRU achieves the best F1 at its validation-selected threshold, but the snapshot GCN slightly exceeds it in AUROC and AUPRC. This finding is useful rather than inconvenient: graph structure appears essential in the benchmark, whereas temporal complexity should be justified by the data rather than assumed.

For practical deployment, the model must be supported by point-in-time data engineering, effective-dated graph construction, probability calibration, explanation, data governance, and continuous drift monitoring. The strongest research opportunity is to validate the approach on real multi-tier supply networks and integrate prediction with digital-twin simulation and mitigation optimization. In that setting, graph neural networks can move beyond static risk scoring toward a continuously updated early-warning capability that detects where disruption is likely to emerge, how it may propagate, and which parts of the network deserve attention first.

### **9.1. Data Availability and Reproducibility Statement**

The empirical results in this manuscript are based on a reproducible semi-synthetic benchmark created for methodological evaluation. No proprietary company dataset is represented as real data. The benchmark contains a four-tier directed supply network, supplier-risk variables, time-varying logistics variables, and disruption labels generated by a documented nonlinear cascade process. A journal submission should include the generator code, random seed, model configuration, and metric script as supplementary material. External validation using real operational data remains necessary before deployment claims can be generalized.

### **9.2. Conflict of Interest**

The authors should disclose any financial, commercial, or organizational relationships relevant to supplier-risk data, logistics platforms, or analytics software before submission. No conflict-of-interest status is assumed in this manuscript; the authors should insert the declaration required by the target journal before submission.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest.

**Publisher's Note:** All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers.

### **References**

- Garvey, M. D., Carnovale, S., and Yeniyurt, S. "An Analytical Framework for Supply Network Risk Propagation: A Bayesian Network Approach." *European Journal of Operational Research* 243, no. 2 (2015): 618-627. <https://doi.org/10.1016/j.ejor.2014.10.034>
- Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. "On Calibration of Modern Neural Networks." *Proceedings of the 34th International Conference on Machine Learning* 70 (2017): 1321-1330.
- Hamilton, W. L., Ying, Z., and Leskovec, J. "Inductive Representation Learning on Large Graphs." *Advances in Neural Information Processing Systems* 30 (2017).
- Hosseini, S., Ivanov, D., and Dolgui, A. "Review of Quantitative Methods for Supply Chain Resilience Analysis." *Transportation Research Part E: Logistics and Transportation Review* 125 (2019): 285-307. <https://doi.org/10.1016/j.tre.2019.03.001>
- Hosseini, S., Tajik, N., Ivanov, D., Sarder, M. D., Barker, K., and Al Khaled, A. "Resilient Supplier Selection and Optimal Order Allocation under Disruption Risks." *International Journal of Production Economics* 213 (2019): 124-137. <https://doi.org/10.1016/j.ijpe.2019.03.018>.
- Ivanov, D. "New Disruption Risk Management Perspectives in Supply Chains: Digital Twins, the Ripple Effect, and Resilience." *IFAC-PapersOnLine* 52, no. 13 (2019): 337-342. <https://doi.org/10.1016/j.ifacol.2019.11.138>.
- Ivanov, D., and Dolgui, A. "A Digital Supply Chain Twin for Managing the Disruption Risks and Resilience in the Era of Industry 4.0." *Production Planning & Control* 32, no. 9 (2021): 775-788. <https://doi.org/10.1080/09537287.2020.1768450>
- Ivanov, D., and Dolgui, A. "Viability of Intertwined Supply Networks: Extending the Supply Chain Resilience Angles Towards Survivability. A Position Paper Motivated by COVID-19 Outbreak." *International Journal of Production Research* 58, no. 10 (2020): 2904-2915. <https://doi.org/10.1080/00207543.2020.1750727>
- Ivanov, D., Sokolov, B., and Dolgui, A. "The Ripple Effect in Supply Chains: Trade-Off 'Efficiency-Flexibility-Resilience' in Disruption Management." *International Journal of Production Research* 52, no. 7 (2014): 2154-2172. <https://doi.org/10.1080/00207543.2013.858836>

- Kipf, T. N., and Welling, M. "Semi-Supervised Classification with Graph Convolutional Networks." International Conference on Learning Representations, 2017. arXiv:1609.02907.
- Ojha, R., Ghadge, A., Tiwari, M. K., and Bititci, U. S. "Bayesian Network Modelling for Supply Chain Risk Propagation." International Journal of Production Research 56, no. 17 (2018): 5795-5819. <https://doi.org/10.1080/00207543.2018.1467059>
- ossi, E., Chamberlain, B., Frasca, F., Eynard, D., Monti, F., and Bronstein, M. "Temporal Graph Networks for Deep Learning on Dynamic Graphs." arXiv:2006.10637, 2020.
- Pareja, A., Domeniconi, G., Chen, J., Ma, T., Suzumura, T., Kanezashi, H., Kaler, T., Schardl, T. B., and Leiserson, C. E. "EvolveGCN: Evolving Graph Convolutional Networks for Dynamic Graphs." Proceedings of the AAAI Conference on Artificial Intelligence 34, no. 4 (2020): 5363-5370. <https://doi.org/10.1609/aaai.v34i04.5984>.
- Saito, T., and Rehmsmeier, M. "The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets." PLOS ONE 10, no. 3 (2015): e0118432. <https://doi.org/10.1371/journal.pone.0118432>.
- Taj, S., Imran, A. S., Kastrati, Z., Daudpota, S. M., Memon, R. A., and Ahmed, J. "IoT-Based Supply Chain Management: A Systematic Literature Review." Internet of Things 24 (2023): 100982. <https://doi.org/10.1016/j.iot.2023.100982>
- Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. "Graph Attention Networks." International Conference on Learning Representations, 2018. arXiv:1710.10903.
- Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Yu, P. S. "A Comprehensive Survey on Graph Neural Networks." IEEE Transactions on Neural Networks and Learning Systems 32, no. 1 (2021): 4-24. <https://doi.org/10.1109/TNNLS.2020.2978386>.
- Ying, Z., Bourgeois, D., You, J., Zitnik, M., and Leskovec, J. "GNNExplainer: Generating Explanations for Graph Neural Networks." Advances in Neural Information Processing Systems 32 (2019).
- Yu, B., Yin, H., and Zhu, Z. "Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting." Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (2018): 3634-3640. <https://doi.org/10.24963/ijcai.2018/505>.
- Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., and Sun, M. "Graph Neural Networks: A Review of Methods and Applications." AI Open 1 (2020): 57-81. <https://doi.org/10.1016/j.aiopen.2021.01.001>.