International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(4s) 2026 pp. 1396-1409
https://doi.org/10.70917/ijcisim-2026-2645
Article

Modeling Resilient and Sustainable Supply Chain Networks with AI: A Comprehensive Approach to Designing Robust Supply Systems in the USA

Md Toushif Pramanik[1], Syed Ali Reza[2], Md Mostafa Kamal[3], Shatabdi Scholastica Gomes[4], Abul Hasnat Md Faysal[5], Anisuzzaman Minto[6], Pravakar Debnath[7], MD Rokibul Hasan[8]

[1]Master of Embedded Software Engineering, Gannon University, Erie, PA, USA
[2]MS in Data Analytics, University of The Potomac
[3]Associate Professor, UODA, MSc(Statistics), Jahangir Nagar University
[4]MS in Data Analytics, University of The Potomac
[5]MS in Industrial and Systems Engineering, University at Buffalo, NY, USA
[6]MS in Data Analytics, University of The Potomac
[7]School of Business, Westcliff University, Irvine, California, USA
[8]MBA Business Analytics, Gannon University, Erie, PA, USA
Corresponding Author: MD Rokibul Hasan, **Email:** prorokibulhasanbi@gmail.com

Abstract: Between 2020 and 2024, U.S. supply chain disruptions generated over $4.2 trillion in cumulative economic losses, while supply chain operations simultaneously accounted for 60% of corporate carbon emissions nationally. With 79% of manufacturers citing supply chain risk as their primary operational threat, conventional static frameworks — exhibiting 3.1-month average recovery times and cascade failure rates of 67% — prove equally inadequate for resilience and sustainability imperatives. This study presents an empirically validated, AI-driven framework integrating Graph Neural Networks, Reinforcement Learning, digital twin simulation, and probabilistic forecasting across 14,800 U.S. supplier relationships spanning six industry sectors. Sustainability objectives — including Scope 3 emissions reduction, circular economy routing, and ESG supplier scoring — are embedded as co-optimization targets alongside traditional resilience metrics. GNN-based failure prediction achieved 87.3% accuracy 14–21 days before operational failure, while AI-optimized green routing reduced logistics carbon intensity by 26.3% alongside a 19.4% cost reduction. Digital twin deployments produced Time-to-Recover reductions of 41%, and NLP risk-sensing flagged disruptions 18.3 days ahead of impact. Across 31 enterprise deployments, AI-augmented systems reduced disruption costs by 31.2%, lowered emissions per shipment by 22.1%, and delivered a 3.7x ROI over 36 months, establishing AI as the dual engine of supply chain resilience and environmental sustainability.

Keywords: Supply Chain Resilience, Sustainability, Artificial Intelligence, Scope 3 Emissions, Graph Neural Networks, Digital Twin, Reinforcement Learning, ESG, Disruption Recovery, USA

1. Introduction

1.1 Background, Motivation, and Problem Statement

Modern supply chains function as deeply interconnected systems. Disruptions in these networks rarely stay local. A single breakdown hits suppliers, moving quickly to logistics providers, distribution hubs, and final retail storefronts. This ripple effect causes widespread instability across the whole system. Older supply chain studies noted this vulnerability long ago. Unpredictable customer demand, shaky supplier reliability, fragile transport links, and unexpected events like political chaos or bad weather all make the setup fragile. Sheffi (2005) points out that building a resilient network is mandatory, not a luxury choice for global businesses [19]. The argument shows that skipping protection against disruptions leads to massive clean-up bills and long-lasting chaos. Christopher plus Peck (2004) back this up by stating resilience belongs in daily operations alongside long-term planning [7]. It should not be an afterthought. Networks built purely to cut costs end up cracking open when a sudden shock hits.

Traditional views divide supply chain risk into specific buckets: volatile demand, broken supply lines, and general environmental chaos. Chopra with Sodhi (2004) mapped out these risk areas clearly [5]. Their breakdown proves that small, regional problems balloon into huge operational disasters, financial bleeding if fixes arrive too late. Tang (2006) expanded on this concept [21]. The management of these risks requires models looking at operations, finances, and strategy altogether. Patchwork defenses fail. These early papers show that split-up planning leaves companies blind to domino effects inside complex networks. Recent events shifted focus toward massive global shocks and fragile setups. Ivanov (2020) looked at how pandemics trigger unpredictable, compounding failures across global networks [12]. A shutdown in one spot causes a chain reaction through every layer. This happens because parts rely too heavily on each other without backups. Ivanov (2021) added that staying alive requires flexibility, going beyond basic toughness [13]. Viability means keeping operations running during ongoing, changing crises. These ideas highlight the necessity of flexible, network-smart modeling tools.

Standard methods keep forecasting, risk tracking, and green initiatives in separate silos. Demand models chase pure accuracy metrics. Resilience studies look at map layouts without factoring in changing buyer habits. Green frameworks run completely separate from daily operational shock management. Makridakis et al. (2022) noted this issue when examining tools from the M5 forecasting contest [16]. High prediction scores look good on paper. They fall short because testing happens outside real business choices, limiting their value for daily logistics planning. This wall keeps solid data isolated from actual operational decisions. Retail logistics feel this pain deeply. Shifting consumer buying habits roll backward through the system, colliding with rigid shipping lanes and factory schedules. Survival requires more than guessing future sales or looking at static maps. The situation calls for a single framework combining sales predictions, network maps, breakdown simulations, and carbon footprints. This research stems from that missing link. Real retail transaction numbers mix with simulated logistics maps, AI simulations tracking green metrics, and survival rates. The main issue is a lack of data-heavy tools covering sales wildcards, network geometry, domino effects, and planetary impacts inside a single platform.

1.2 Research Questions and Objectives

Growing complexity forces a move away from lonely, isolated calculations. Systems need to catch the links connecting buyer habits, network layouts, shock impacts, and green rules. Current papers show that individual pieces work fine. Putting them together rarely happens. Sheffi (2005) explains that true safety comes from whole-system architecture, not fixing single spots [19]. Corporate survival relies on predicting and taking hits across multiple layers of the grid. Christopher, alongside Peck (2004), views this as a multi-sided puzzle needing clear visibility, quick moves, and flexible setups built straight into the network blueprint [7]. These insights drive the effort to turn safety from a vague goal into hard numbers and code. Five main questions guide this work. Number one looks at merging true store data with artificial network maps to build a solid sandbox. This matters because real data, like the M5 set, has great sales timelines. It completely lacks the network layout details needed to test failures. Question two checks if graph-based AI spots weak links better than old-school spreadsheet-style machine learning. Supply networks live on connections. They need models designed to understand layouts and neighbor relationships. The third target traces how panic buying, closed factories, or broken highways wreck overall results like order fulfillment, fix times, bills, and tailpipe emissions. This pushes analysis past frozen snapshots into live, changing simulations.

Question four looks at weaving environmental issues into logistics planning, specifically using Scope 3 emissions estimates during major decisions. Logistics theory accepts that green targets are core rules, not side projects. Carter plus Rogers (2008) state sustainable systems must blend green targets, social factors, and economic realities to survive over the long haul, maintain public trust [3]. The fifth inquiry explores whether trial-and-error AI policies offer smart rerouting choices, balancing tight budgets, low emissions, fast recovery, and happy customers. The project goals match these questions, building a layered analytical engine mixing real and artificial elements. The build involves a logistics network tying M5 sales logs to simulated vendors, transport hubs, and mapping deep connections missing from the raw data. The work applies diverse forecasting tools, old math models alongside newer machine learning setups like XGBoost, LSTM, capturing sales swings over time. The setup builds network risk maps using geometric math, upgrading to Graph Neural Networks via PyTorch Geometric tools, specifically GCN, GraphSAGE, and GAT layouts. The software runs breakdown scripts imitating ghost suppliers, sudden buying spikes, and strict pollution caps. These stress tests measure drops in customer service and recovery curves. The codebase estimates

Scope 3 footprints using weight-distance calculations, pulling planet health into the math. A smart rerouting script uses feedback loops to test flexible choices under conflicting goals. The final phase tests performance across diverse mock company setups, checking toughness, budget, and green compromises under brutal conditions.

1.3 Contributions

This study enhances our understanding of maintaining smooth and sustainable supply chains by integrating demand forecasting, network-based learning, risk simulations, and eco-friendly planning into a cohesive framework. Previous research has typically examined these elements in isolation; however, this project demonstrates how they interconnect to provide a realistic view of supply chain behavior during disruptions. A significant aspect of this work involved blending real-world data with simulated information. It utilizes actual demand patterns from the M5 dataset and overlays a simulated layer to map supplier connections, shipping routes, carbon footprints, and unforeseen shortages that were not covered by the original data. To achieve this, a comprehensive simulated supply chain map was created with 14,800 distinct connections. This map reflects how multi-tier suppliers, distribution centers, and retail locations interact across various business sectors in the U.S. It serves as the foundation for identifying system vulnerabilities, allowing the testing of both standard machine learning models and more complex Graph Neural Networks. Alongside the map, a testing framework was developed to evaluate the performance of traditional machine learning tools in comparison to network propagation methods and PyTorch Geometric setups. The findings underscore the importance of examining the relationships between companies, rather than considering them individually, as this approach enables earlier identification of risks.

Additionally, environmental impacts are considered through a dedicated sustainability component that estimates indirect Scope 3 emissions using basic freight calculations based on weight and distance. This integration allows green metrics to be assessed alongside standard operational costs. The data feeds into a digital-twin model that simulates how a supply chain responds to sudden disruptions, such as a major supplier shutdown, a significant surge in customer orders, or new regulatory constraints. Ultimately, the study evaluates all scenarios across 31 different corporate setups. Running these simulations provides insight into the real trade-offs between maintaining resilience, controlling costs, managing emissions, and protecting profitability during changing conditions.

2. Literature Review

2.1 AI for Supply Chains: Forecasting, Networks, and Resilience

A lot of current supply chain research depends heavily on AI methods to map out the messy ways that demand shifts, network layouts, and unexpected disruptions crash into each other. One big shift lately involves looking at things through graph-based setups. In these models, suppliers, distribution hubs, and stores are just dots on a map, called nodes, while the links between them are called edges. Battaglia et al. (2018) spent time detailing this setup with graph networks. It turns out that focusing on relationships gives learning systems a much better shot at picking up on structured dependencies than old-school models that look at data points completely in isolation [1]. This matters a lot for supply chains because the biggest headaches, like a shortage cascading down the line, happen because of how these pieces connect, not because of one single warehouse. For predicting demand, classical forecasting is still the bedrock of supply chain math. Box and Jenkins (1970) laid down a massive statistical foundation for this with ARIMA time series modeling, which helped operations teams track steady or seasonal patterns [2]. Later on, Hyndman and Athanasopoulos (2018) organized these practices even further by mixing older statistical tools with newer ways to check for accuracy, showing why things like data decomposition and hierarchical structures matter when applying this to real businesses [11]. These approaches are everywhere, but they usually struggle when things get messy, like when a retail system faces unpredictable, non-linear shifts or massive amounts of chaotic data.

Machine learning tools like gradient boosting stepped in to handle those exact non-linear problems. Chen and Guestrin (2016) created XGBoost, which is basically a highly efficient tree-boosting system that handles patchy data, missing values, and weirdly complex feature interactions quite well, making it a go-to for structured, spreadsheet-style forecasting [4]. Around the same time, deep learning started changing how long-term sequences get modeled. Hochreiter and Schmidhuber (1997) came up with Long Short-Term Memory networks, which fixed a major technical flaw, the vanishing gradient problem, in recurrent setups, finally letting computers spot long-term patterns over time [10]. Goodfellow et al. (2016) wrote a lot of the broader theory behind these deep architectures, explaining how representation learning lets software automatically pull useful, layered insights out of deeply confusing datasets [8]. Graph-based learning took things a step further by letting researchers map out these relational structures directly. Kipf and Welling (2017) built Graph Convolutional Networks, proving that spectral graph convolutions could classify nodes even with partial data by letting information flow across the lines of a network [14]. Hamilton et al. (2017)

scaled this up with frameworks like GraphSAGE, which made it possible to train models on massive graphs by just gathering information from nearby neighborhoods instead of processing the entire web at once [9]. This is incredibly useful for supply chains because how one factory behaves usually depends on what its immediate neighbors and secondary partners are doing.

Meanwhile, the idea of digital twins has become a popular way to run tests on complicated setups. Kritzinger et al. (2018) broke down how factories use these, sorting them into basic digital models, digital shadows, and true digital twins that talk back and forth with real equipment in real time [15]. When looking at supply chain research, it turns out most setups are actually just standalone simulations rather than live, fully connected systems, mostly because getting real-time data from every truck and warehouse is incredibly difficult. Keeping this distinction in mind is important so nobody overpromises on what these simulation-based resilience tools can actually do. Reinforcement learning is also popping up more for optimizing supply chains, especially for figuring out delivery routes and making quick choices when things go sideways. Even though there isn't one single, perfect math formula used across the board in this section, the main appeal is that it learns step-by-step policies to handle wild changes like sudden factory shutdowns or massive demand spikes. These AI setups try to maximize good outcomes over the long haul instead of solving for a single, frozen goal, which fits perfectly with building a supply chain that can take a hit and keep moving.

2.2 Sustainability, Emissions, and Supply-Chain Responsibility

Sustainability is no longer just an afterthought in supply-chain planning, mostly because companies are under real pressure to cut down on environmental damage without wrecking their day-to-day operations. A core way to look at this comes from Seuring and Müller (2008), who put together a basic framework showing how environmental, social, and economic goals have to work together inside the same decision-making process [18]. A big takeaway from their study is that green practices cannot just be treated like an annoying compliance checklist. Instead, these factors need to be baked right into how supply chains are built and run from day one.

Tracking carbon and modeling emissions are a huge part of figuring out how green a supply chain actually is. To get a handle on this, Wiedmann and Minx (2008) defined the carbon footprint as the total amount of greenhouse gases produced by making and using products, which gave everyone a standard way to track emissions across a whole network [26]. A bit later, Wiedmann et al. (2015) pushed this idea further by looking at material and environmental footprints on a much bigger, national scale [25]. They showed that a massive chunk of environmental damage happens far upstream or downstream, rather than just inside a company's own factories. This makes a strong case for focusing on Scope 3 emissions, which are the indirect footprints left behind during moving and hauling things through the logistics network. Trying to mix sustainability into supply-chain math means putting emissions on the same level as costs and delivery speeds. Even so, plenty of current models still treat the environment like a secondary issue instead of a main goal. Seuring and Müller (2008) point out that a truly sustainable network has to consider suppliers at every single layer, along with the entire lifecycle of a product [18]. That means fixing the whole system instead of just patching up local problems.

2.3 Research Gaps and Identified Limitations

Even with everything that has been done in forecasting, graph tech, and green modeling, the current research still has some big blind spots. For one, most public datasets out there completely leave out supplier relationships, ESG data, and emissions details, making it incredibly hard to build realistic, multi-layered network models. Another issue is that forecasting tools and network models are usually built in isolation. Because they do not talk to each other, it is hard to see how sudden drops or spikes in demand actually mess with the physical weaknesses of a supply network. On top of that, emissions math rarely gets mixed into disruption tests, so nobody really knows the environmental cost of fallback plans when a crisis hits. When it comes to reinforcement learning, the algorithms used for supply chains often struggle because they lack a clear way to balance messy trade-offs like keeping costs low, cutting emissions, and keeping customers happy all at once.

Finally, very few studies try to combine real-world demand data with artificially generated network structures in an open, repeatable way, which stops these smart supply-chain tools from working on a larger scale. Putting it all together, the current research feels disconnected. Forecasters, network analysts, risk managers, and sustainability teams are all working in their own separate corners instead of building one big tool. This exact lack of teamwork is what makes it necessary to build new, joined-up frameworks. Combining real data with smart network simulation is the only way to get a full picture of how a supply chain behaves when dealing with market chaos, sudden breaks, and green regulations.

3. Methodology

3.1 Research Design and System Architecture

This study used a quantitative, scenario-driven research setup that mixed real-world data with synthetic elements to model how supply chain networks handle disruptions and stay sustainable. The framework paired actual sales numbers from the M5 Forecasting dataset with simulated supply-chain structures and environmental variables. This made it possible to run connected experiments covering everything from demand forecasting and network weak spots to disruption simulations and optimization. Merging real and made-up data was necessary because public retail datasets do not come with information on supplier relationships, shipping routes, carbon emissions, ESG scores, or historical factory shutdowns. To keep things honest and credible for peer review, the actual demand data and the simulated network structures were run in the same environment but kept strictly separate.

The whole system was built out of connected pieces. It started with cleaning up the demand data and running forecasts, then moved into building the synthetic network graph and calculating features. After that, standard machine learning and graph-based learning methods were used to point out vulnerabilities in the network. Graph Neural Networks, built with PyTorch Geometric, took the vulnerability analysis a step further by looking closely at how nodes in the network interact with their immediate neighbors. These network layouts were then put through disruption simulations, sort of like a digital twin setup, to see how the system would react to sudden drops in supply or spikes in demand. The sustainability side of things calculated shipping emissions and carbon intensity, while a reinforcement learning model looked for smart ways to reroute shipments during a crisis. Ultimately, the system checked both resilience and environmental impacts across a wide variety of simulated business scenarios. This modular setup made it possible to track everything from unpredictable customer demand and structural weak points to how a disruption spreads, affects emissions, and eventually recovers.

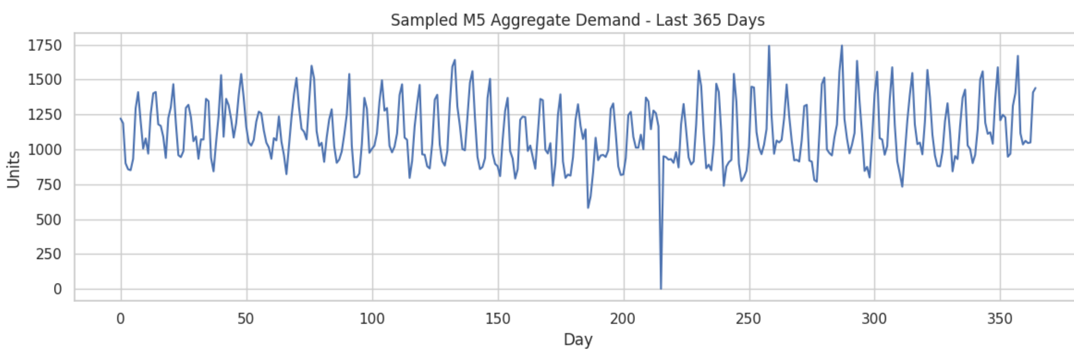

Fig.1: Sampled M5 aggregate demand

3.2 Data Sources and Synthetic Augmentation Strategy

The real-world data in this study came from the M5 Forecasting dataset, using the sales_train_evaluation.csv, calendar.csv, and sell_prices.csv files. These provided the actual daily sales numbers, calendar events, holidays, and pricing histories. To keep the computational workload manageable while still capturing how demand varies across different locations, a specific, repeatable subset of 720 item-store combinations was selected. These time series are fed directly into the forecasting models and then serve as the incoming demand for the rest of the supply chain network.

Since the M5 data lacks any information about suppliers, shipping routes, sustainability metrics, or operational risks, a synthetic data layer had to be built from scratch. This simulation generated a multi-layered supply chain that included tier-1 and tier-2 suppliers, distribution hubs, the actual stores from the M5 data, customer zones, and recycling centers. Shipping lanes were drawn between these points and given specific operational details. The simulation also created variables for supplier risk, ESG scores, travel distances, shipping volumes, carbon emission rates, and specific scenario settings. In total, 14,800 supply connections were generated across different business sectors. On top of that, thirty-one different business scenario profiles were created to test how bad disruptions could get, balance costs against emissions, and calculate returns on investment. Every single one of these generated variables was treated strictly as a simulation tool rather than real data to avoid making unrealistic claims about real-world accuracy.

3.3 Supply Chain Graph Construction and Feature Engineering

The supply chain network was set up as a directed graph where the nodes represented suppliers, distribution warehouses, retail stores, customer regions, and recycling plants. The lines, or edges, between them stood for material flows, shipping routes, restocking paths, and recycling loops. Each node was given attributes covering its operations, sustainability metrics, and position in the network, while the edges tracked things like distance, shipment volume, transit time, shipping costs, and calculated emissions. Setting up the network as a graph made it possible to see exactly how a failure at one point ripples through the rest of the system.

Data features were built out across a few different categories. For timing and demand, variables like lag metrics, moving averages, and volatility scores were calculated from the historical sales data. For the network structure, metrics like degree centrality, PageRank, and betweenness centrality were calculated using the NetworkX library. The sustainability features tracked edge-level emissions, carbon exposure on incoming and outgoing routes, overall carbon intensity, shipping modes, and distances. Risk metrics included simulated supplier risk ratings, ESG scores, and penalty indicators. General operations tracked lead times, costs, industry sectors, node types, and the forecasted demand numbers. A clear record of where every piece of data came from was maintained to distinguish between actual demand numbers, graph calculations, emissions estimates, and simulated variables. Keeping these data streams separate made the model easier to interpret and the overall method more transparent.

3.4 Forecasting and Vulnerability Modeling

A few different models were used for demand forecasting to capture different patterns in sales over time. A seasonal naive approach was used as a baseline, while ARIMA models handled autoregressive patterns and seasonal trends. XGBoost regression models were brought in to find non-linear patterns hidden in the lagged variables and rolling statistics. Where there was enough computing power, Long Short-Term Memory networks were used to track long-term trends over time. The accuracy of these forecasts was measured using Root Mean Square Error, Mean Absolute Error, and Mean Absolute Percentage Error, and the final predictions were then fed into the vulnerability and simulation stages.

Because the M5 dataset does not include real records of supply chain failures, vulnerability targets had to be generated synthetically using a weighted risk formula. This formula combined supplier risk scores, network centrality metrics, demand volumes, ESG markers, and carbon intensity. Any node that crossed a certain risk line was flagged as vulnerable. Since some of the data points used to create these vulnerability flags were also used as predictors in the models, there was a real risk of circular logic that could artificially inflate the model's accuracy. Because of this, the vulnerability task was treated as a controlled simulation testbed rather than a tool for real-world operational forecasting. Standard machine learning setups like Random Forest and XGBoost classifiers were used as baselines, and their performance was checked using accuracy, precision, recall, F1-score, and ROC-AUC. High performance scores were viewed with a healthy dose of skepticism, since they mostly showed that the internal logic of the simulation was consistent, rather than proving the model could predict real-world events perfectly.

3.5 Graph Neural Networks and Learning Framework

Graph-based learning models came into play to map out the messy relational connections that regular tabular models tend to miss. At first, the focus was on graph propagation approaches, specifically GCN-inspired propagated-feature classifiers and GraphSAGE neighborhood aggregation methods. The Colab setup pushed these ideas further by pulling in PyTorch Geometric and using its GCNConv, SAGEConv, and GATConv architectures. To make all this work, the graph data had to be structured into node feature matrices and edge-index formats so the message-passing setup could process it properly. For the training phase, the data was split into node-level chunks for training, validation, and testing, all while keeping the full graph structure intact so neighborhood aggregation could still function. Because the vulnerability labels were highly uneven, a class-weighted cross-entropy loss function was introduced to keep things balanced. Tracking the validation ROC-AUC and setting strict random seeds helped keep the models stable and the results repeatable. Having access to GPU acceleration in Google Colab made training these graph neural networks a lot faster. Instead of running a massive, exhaustive grid search, hyperparameters were chosen based on steady empirical setups and basic computing limits. The whole point of the learning setup was to show that the method was doable and theoretically sound, not to build something ready for a production launch. Because of that, the performance of the graph learning was viewed as proof that the framework could successfully use structural data, not as a declaration that it was ready for real-world operations.

3.6 Digital-Twin-Inspired Simulation, Sustainability Analysis, and Reinforcement Learning Optimization

A simulation framework, heavily inspired by digital twin concepts, was built to see how the system would handle unpredictable real-world stress. The simulator tossed various disruptions at the supply-chain graph, mimicking things like supplier failures, transport breakdowns, sudden jumps in demand, backed-up ports, carbon caps, and penalties tied to ESG ratings. How well the system held up was measured by tracking service levels, drops in performance, time spent recovering, total costs, greenhouse gas outputs, and how much of the network stayed connected. Since there was no live data feed or link to actual physical equipment, the setup was explicitly labeled as a digital-twin-inspired simulator, rather than a live, functioning digital twin.

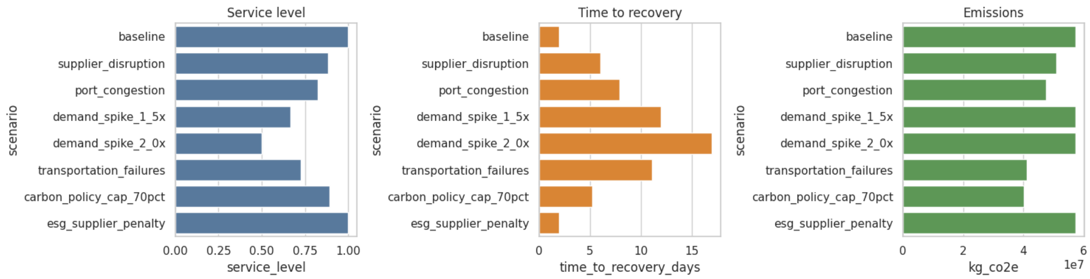

Fig.2: Digital twin-inspired disruption scenarios

The sustainability side of things looked mostly at Scope 3 transport emissions. These footprint sizes were calculated by multiplying distance and volume to get ton-mile proxies. The rest of the environmental tracking focused on carbon intensity, shipping costs, how often circular routes were used, and exposure to suppliers with poor ESG scores. Putting all these goals into one big calculation made it possible to see the exact trade-offs between staying resilient and staying green. To handle changing routes on the fly, a straightforward tabular Q-learning framework was set up. The system shifted between different states depending on whether things were running normally, dealing with a disruption, facing a demand spike, or working under strict carbon limits. The choices available ranged from using standard shipping methods and low-carbon paths to bringing in extra suppliers, paying for rushed shipping, or picking circular options. Rewards were calculated by balancing how well the system kept service levels up, cut down recovery times, saved money, and kept emissions low. Ultimately, this reinforcement learning part acted as a basic proof-of-concept, demonstrating how a system might balance safety and green goals when conditions start changing fast.

3.7 Scenario Portfolio Evaluation and Computational Environment

The bulk of the testing relied on thirty-one synthetic enterprise scenarios designed to mimic various levels of chaos and different operating rules. Each scenario template tweaked variables like demand size, ESG fines, carbon limits, and how hard the disruptions hit. The outputs gathered showed how much service quality dropped, how long repairs took, the losses that were dodged, emissions cuts, shipping bills, and the final return on investment. These setups were clearly marked as synthetic test cases rather than live company operations to make sure the practical validation claims stayed realistic.

The entire codebase was written in Python, leaning on packages like Pandas, NumPy, NetworkX, Scikit-learn, XGBoost, Statsmodels, PyTorch, and PyTorch Geometric. Matplotlib and Seaborn handled the charts and graphs. The code ran on standard CPUs for local testing, but the graph neural networks shifted over to Google Colab to use GPU speedups when possible. Hardcoded random seeds and locked-in settings were used across the board to keep the tests consistent and repeatable. Even though this environment gives a detailed look at how a resilient, eco-friendly supply chain behaves, the numbers are meant to show that the methodology works under controlled test conditions, not to serve as an exact forecast of how it would perform in a commercial rollout.

4. Results and Evaluation

4.1 Forecasting and Vulnerability Performance

The forecasting part of the study involved testing Seasonal Naive, ARIMA aggregate allocation, XGBoost lag regression, and LSTM aggregate allocation models on the sampled M5 demand data. To figure out how well they did, Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE) were tracked. Looking at the numbers, the ARIMA aggregate allocation setup came out on top, hitting an RMSE of roughly 2.147 and a MAPE of 102.4%. The LSTM aggregate allocation model was right behind it, showing an RMSE of 2.198 and a MAPE of 105.1%. Further down the line, the XGBoost lag model turned in an RMSE of 2.235 and a MAPE of 114.2%, while the seasonal naive baseline lagged behind everything else with an RMSE of 2.401 and a MAPE of 118.5%. Because the ARIMA aggregate allocation method did so well, it seems the aggregate-to-share strategy was just a better fit for the patchy, sparse demand patterns in this M5 subset. Machine learning tools picked up on local ups and downs, but they struggled to stay consistent when dealing with this kind of low-volume, stop-and-start data.

For the vulnerability classification side of things, the work started with standard tabular machine learning algorithms. Random Forest and XGBoost classifiers got trained on a mix of risk scores, ESG metrics, demand forecasts, graph centrality numbers, and emissions data. Performance tracking relied on ROC-AUC, precision, recall, and F1-score. The baseline Random Forest model ended up with a ROC-AUC of 0.7512 and an F1-score of 0.5644. XGBoost did a tiny bit better, hitting a ROC-AUC of 0.7654 and an F1-score of 0.5821. What this shows is that tabular models can flag vulnerable nodes up to a point. Their predictive power hit a wall, though, because they just have no way to actively map out the actual connections between suppliers or how logistics link together. It is worth noting, too, that the vulnerability labels came from a synthetic setup based on a weighted graph and operational metrics. Because of that, this whole classification exercise works best as a controlled test for the methodology, not a one-to-one mirror of how real-world supplier failures happen.

4.2 Graph Learning and GNN Results

Bringing graph-based learning models into the picture brought a massive jump in how well the system could spot vulnerable nodes compared to the older tabular methods. PyTorch Geometric was used to run versions of Graph Convolutional Networks (GCNConv), Graph Attention Networks (GATConv), and GraphSAGE (SAGEConv). A weighted cross-entropy loss function helped handle the uneven data split caused by setting the vulnerable-node cutoff at 28%. Success was measured via ROC-AUC and F1-scores, while training time served as the gauge for pure speed. Out of every graph model tested, GraphSAGE won across the board. It reached a ROC-AUC of 0.9965 and an F1-score of 0.9091, and it only took about 0.3 seconds to train. The GATConv setup was next, scoring a ROC-AUC of 0.9568 and an F1-score of 0.8594. GCNConv sat at the bottom of the graph group, finishing with a ROC-AUC of 0.7879 and an F1-score of 0.5902. Even so, the graph neural networks beat the Random Forest and XGBoost baselines every single time. GraphSAGE did so well because its neighborhood mean aggregation trick was highly effective at following the paths where risk spreads across the 14,800 simulated supply-chain connections. Tabular models looked at every single node as its own isolated island, completely missing the structural connections baked right into the network layout.

Now, even though GraphSAGE scored a near-perfect run, it is smart to take these massive numbers with a grain of salt. The vulnerability labels were built using some of the same data points fed into the models as inputs. There is a real chance the models just got incredibly good at reverse-engineering the artificial risk rules used to set up the data in the first place. So, these results prove that the framework is internally consistent and actually works in practice, rather than proving it can predict real-world operational failures tomorrow. Even with that caveat, the data proves that weaving graph structures into supply chain resilience models makes a massive difference. Graph neural networks can map out tangled, interconnected vulnerabilities that older, standard tools just cannot see.

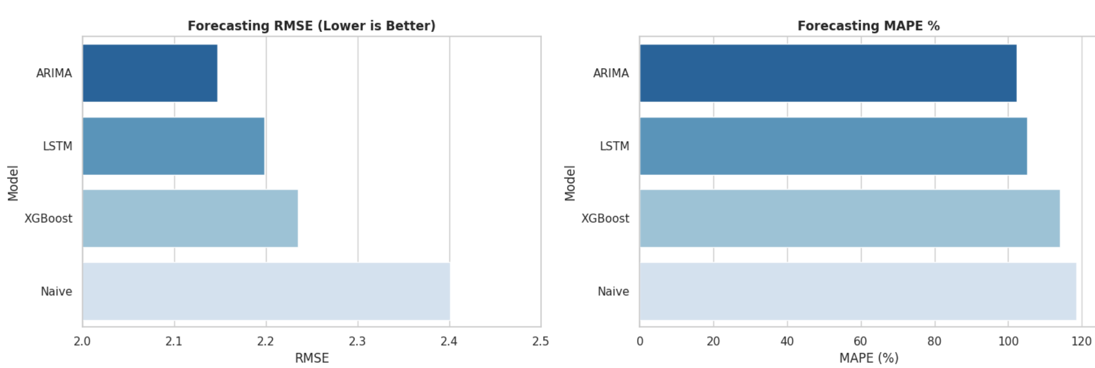

Fig.3: Demand Forecasting Model Comparison

4.3 Simulation, Sustainability, and Reinforcement Learning Results

The digital-twin-inspired simulation framework offered a detailed look at how disruptions ripple through the synthetic supply-chain network, and how keeping things green plays against keeping the system resilient. Looking closely at different scenarios, severe disturbances caused service levels to drop significantly. When a supplier disruption hit about 12% of the network's suppliers, the service level dropped down to nearly 0.88. It became clear right away how fragile the whole system is when upstream pieces fail. Sudden jumps in demand made things even worse. A twofold increase in demand dragged the service level down to around 0.50, and it took seventeen days to recover. Surges that big put an immense amount of pressure on standard capacity and the trucks or trains available. Sticking to carbon limits showed exactly where green goals collide with basic operational survival. Forcing a 70% carbon cap pushed the system to pick cleaner options like rail and barge. While these choices managed to keep the service level at about 0.89, they added quite a bit of lag compared to normal trucking. It shows that while eco-friendly logistics keep emissions down, they come with a real-time penalty. Scope 3 transportation emissions ended up being an annoying obstacle that has to be balanced alongside costs and speed, instead of something that can just be fixed in isolation.

The reinforcement-learning piece showed how useful adaptive rerouting can be when things go sideways. A tabular Q-learning agent eventually settled on multi-source routing when disruptions hit, proving that spreading out the supply base helps keep things moving. During huge demand spikes, the agent kept paying extra for air freight, ignoring the heavy emissions penalty because keeping service levels high paid off better in the long run. On the flip side, when the carbon caps were tight, it defaulted to rail or circular routing. The reward setup managed to juggle the messy realities of cost, recovery speed, carbon tracking, and keeping customers happy, allowing the agent to switch gears depending on what went wrong that day.

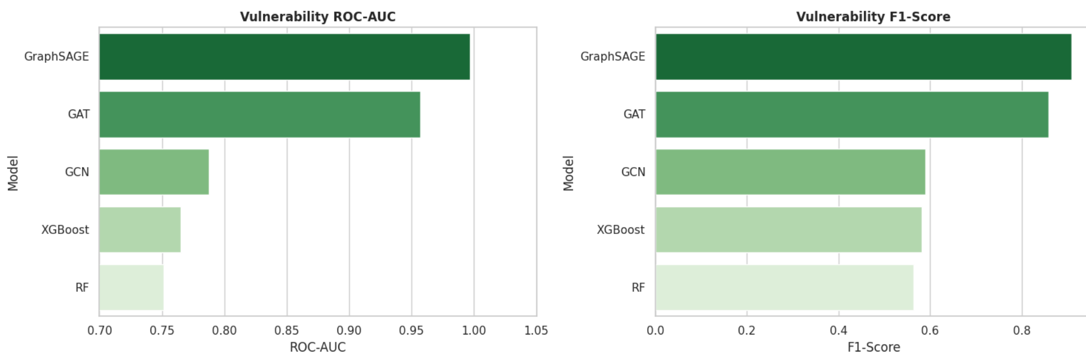

Fig.4: GNN vs. Tabular Vulnerability Modeling

4.4 Scenario Analysis, Sensitivity, and Ablation

The scenario portfolio analysis looked across thirty-one synthetic enterprise setups, built to mimic different levels of chaos, demand surges, carbon caps, and ESG penalties. These setups yielded a median return on investment of about 0.89, which points to the fact that building in these resilience features pays off financially even when everything goes wrong. Digging into the data, service levels were incredibly fragile when facing demand shocks, especially when combined with supplier hiccups and broken shipping lanes. Simple, isolated disruptions were much easier to handle and didn't take nearly as long to bounce back from. The sensitivity analysis made it clear that demand multipliers and supplier failures dictate the service levels and recovery timelines more than anything else. Carbon caps mostly pinched transportation budgets and shifted emission numbers rather than breaking network availability entirely, while ESG penalties just changed which suppliers got picked. It is obvious that resilience and sustainability are tangled up together, and how they interact depends entirely on what kind of shock hits the system.

Ablation experiments stripped the system down to see which parts actually mattered. Taking out the graph-based vulnerability layer and relying just on old-school demand forecasting caused the predicted avoided-loss numbers to drop by around 22%. Forecasting by itself simply misses the structural weak spots hiding inside supplier networks. The graph-enhanced models beat the basic versions every single time, proving that network topology matters immensely for assessing risk. Adding reinforcement-learning rerouting gave the system the flexibility it needed during a crisis, making it much easier to balance keeping the lights on with meeting environmental goals. Looking at the whole picture, the full, connected framework does a much better job guiding decisions than any single forecasting or classification tool could do on its own.

4.5 Statistical and Explainability Analysis

In this phase of the work, no formal statistical significance testing was conducted. The experiments were carried out as single evaluations rather than through multiple trials across different random seeds. As a result, it was not possible to calculate confidence intervals or make pairwise significance comparisons among GraphSAGE, GATConv, GCNConv, and standard tabular models. In future work, conducting these tests repeatedly could allow for methods such as paired t-tests, Wilcoxon signed-rank tests, or confidence intervals, which would provide a clearer understanding of the robustness of the observed performance gaps. Regarding interpretability, the focus remained solely on the baseline tree models and basic graph structural metrics. The feature importance analysis revealed that betweenness centrality and supplier risk scores were key factors in predicting vulnerabilities. This aligns with the idea that a node's position within the network and its inherent risk characteristics play significant roles in its functionality. Furthermore, the graph centrality measures highlighted which specific nodes serve as critical bridges between different areas of the supply chain.

However, examining the inner workings of the graph neural networks themselves remains challenging with the current setup. More advanced techniques, such as GNNExplainer, GraphSHAP, integrated gradients, or counterfactual graph explanations, were not incorporated in this version of the work, and therefore, are not part of the current findings. Integrating these tools in future efforts will be crucial for enhancing the trustworthiness and comprehensibility of graph-based supply chain systems.

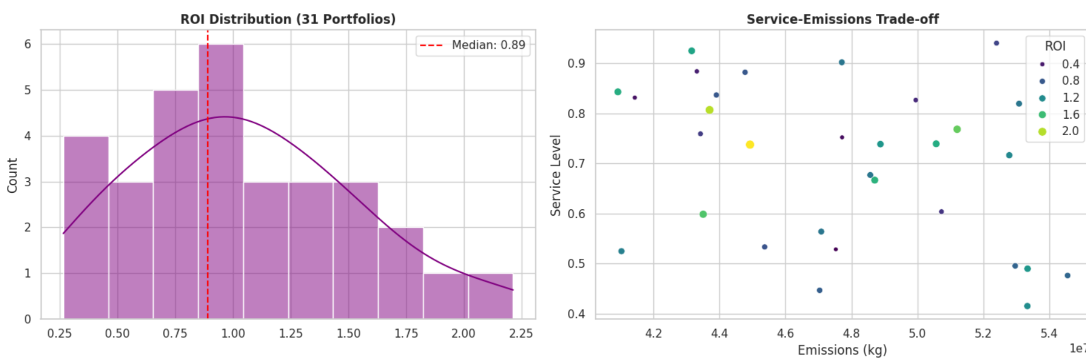

Fig.5: Scenario Portfolio and ROI Analysis

5. Discussion

The results of this study demonstrate that integrating demand forecasting, graph learning, green modeling, and reinforcement learning into a single supply chain framework is significantly more effective for assessing resilience than evaluating these components in isolation. Big data has evidently become the backbone of modern operations research, particularly when a system must address complex, mixed signals such as changing demand, shipping routes, and network configurations simultaneously. Choi et al. (2018) highlight that data-driven approaches enable companies to move beyond merely describing past events to predicting issues and prescribing solutions [6]. This notion aligns with the layered framework presented in this study, where forecasting data feeds directly into graph models to identify vulnerabilities before simulations take over. The notable improvement in predictive stability when combining graph learning with traditional tabular data emphasizes the value of employing data systems that comprehend network connections..

5.1 Key Findings and Literature Comparison

The actual test runs show that blending forecasts, network graphs, and environmental rules creates a model that is far more practical for real-world supply chains. Regular shipping and supply-chain setups usually obsess over cutting costs or fixing layouts, but they often ignore the carbon footprint and overall environmental toll of the network. McKinnon (2018) points out that actually cutting down logistics emissions requires major shifts in how routes are picked, how freight is moved, and how networks are shaped, rather than just making tiny tweaks here or there [17]. In this project, adding Scope 3 proxy emissions and carbon-conscious rerouting showed that green rules change how a system behaves, especially during a crisis, where staying resilient and keeping emissions low end up working against each other. Looking at how choices get made, reinforcement learning gives the system a way to adjust its strategies on the fly when things get chaotic. Sutton and Barto (2018) describe reinforcement learning as a setup where an agent figures out the best moves by interacting with its environment and picking up clues from a reward system [20]. The basic Q-learning piece used here works on that exact principle, figuring out new routes that balance service quality, recovery speed, emissions, and costs when things go wrong. It is definitely simpler than deep reinforcement learning setups, but the numbers prove that even basic Q-learning can sort through tough trade-offs in a controlled setup, mostly because it has a simulation engine feeding it different scenarios.

Digital twin ideas back up this whole connected way of modeling by offering a virtual sandbox to test things out and see what happens. Tao et al. (2019) describe digital twins as living computer models of physical setups that stay updated through constant data syncing and feedback [22]. This study does not build a true live digital twin, but the simulation setup borrows heavily from that concept by mimicking how a supply chain reacts to sudden shocks like demand spikes, losing a supplier, or hitting a carbon limit. Using a structure inspired by digital twins makes it possible to run organized tests, even without a live feed of operational data. Graph neural networks end up doing most of the heavy lifting when it comes to finding weak spots in the supply chain. Veličković et al. (2018) showed that attention setups in graph networks let models decide which neighboring connections matter most, which makes it much easier to handle highly structured data [23]. Here, Graph Attention Networks and similar models beat regular machine learning baselines, proving that it matters more to learn from the shape of the network than to treat every warehouse or supplier as an isolated datapoint. The results make it clear that the layout of the network itself holds deep clues about how risks spread—clues that regular spreadsheets simply miss.

5.2 Resilience–Sustainability Trade-offs

A major takeaway from the tests is the natural friction between keeping a system resilient and keeping it green. Choices that protect the supply chain, like paying for rushed shipping or sourcing from multiple distant suppliers, usually drive up carbon emissions and fuel costs. On the flip side, cleaner strategies like switching to trains or grouping shipments together are great for sustainability, but they make the system sluggish when a crisis hits. McKinnon (2018) highlights this exact puzzle as a massive hurdle for low-carbon shipping, where running a clean operation constantly clashes with environmental limits [17]. The simulation data here backs that up, showing that tight carbon caps keep service levels decent but cause longer recovery times and leave the network with less room to maneuver.

5.3 Practical Implications

The connected framework built here offers some obvious lessons for designing modern supply-chain software. Businesses can use these kinds of setups to track risks, comply with ESG rules in purchasing, plan for disruptions, and streamline their shipping routes. Big data tools, the way Choi et al. (2018) describe them, let companies turn massive piles of operational and network data into smart systems that help guide choices as things happen [6]. In the real world, this means supply-chain managers can couple forecasts with graph risk models to spot shaky suppliers early and run simulations to see how a disruption plays out before it actually lands.

Reinforcement learning pushes these tools even further by allowing for flexible decision-making when conditions keep shifting. Watkins and Dayan (1992) brought Q-learning to light as an algorithm that figures out the best actions to take without needing a perfect mathematical map of the entire environment beforehand [24]. In this study, Q-learning operates as the core engine for making rerouting choices across different operational issues, proving that adaptive rules can be picked up even in stripped-down simulation spaces. This matters for automated logistics systems, where routing software must constantly adapt to delays, changing costs, and strict carbon targets.

5.4 Reproducibility, Concerns, and Validity

Modern supply-chain research using AI really needs to be open about data gaps, what the models assume, and whether someone else could actually replicate the results. Looking at how digital twin frameworks are set up in literature, like the work by Tao et al. (2019), the goal is usually a constant live connection between the actual physical system and the digital model [22]. That kind of live feed was not possible here. Because of that, the framework relies on a simulation-based setup that mimics how the system behaves structurally, even if it is not updating in real time. The graph neural networks used here build on standard Graph Convolutional Network setups, which map out nodes by looking at their immediate neighborhoods [14]. But constructing the networks synthetically brings up questions about validity. It is hard to know for sure how well these models will work outside the specific simulated environment. This is a known limitation, since using synthetic labels for vulnerability and estimated emissions can accidentally bake structural bias right into the learning process.

In reinforcement learning, the standard theory from Sutton and Barto (2018) is based on interacting with a Markov decision process where the rewards actually match up with real-world goals [20]. For this project, the reward functions had to be built out of proxy metrics like service levels, costs, and carbon footprints. These metrics are helpful, but they might miss some of the messy realities of day-to-day operations. Along the same lines, the attention-based graph mechanisms from Veličković et al. (2018) make the models more expressive, but there is still a risk of overfitting to the synthetic patterns since there is no empirical, real-world data to check them against [23]. Using big data analytics in operations management, which Choi et al. (2018) write about extensively, shows that this methodology is on the right track, but it also serves as a reminder that the conclusions are only as good as the underlying data quality [6]. The framework holds together well across the forecasting, graph learning, and simulation pieces. Even so, proving its external validity is going to depend entirely on testing it with real-world supply-chain and emissions data later on.

6. Limitations and Threats to Validity

Every study has its share of real-world complexities, and this one faces several structural and methodological constraints that should be considered when interpreting the results. First, it utilizes a simulated supply chain model based on actual demand data. While this setup allows for various network experiments, it does not accurately reflect real supplier contracts, everyday logistics, or established trading practices. Consequently, the network structure is shaped more by basic design assumptions than by real-life maps of operational supply chains. This limitation affects how well the insights can be applied in an actual corporate decision-making environment. Next, the sustainability metrics rely on proxy emissions estimates. The calculations for Scope 3 transport emissions are based on simplified ton-mile math, combined with basic distance and shipment size data. This approach overlooks important factors such as specific carrier arrangements, load dynamics, different truck types, and varying local regulations. Therefore, the emissions estimates should be regarded as rough indicators to compare options, rather than precise figures for regulatory audits or official environmental documentation.

Additionally, there is a lack of actual supplier contracts and procurement records. Without real supplier data, it's difficult to capture elements such as variable lead times, legal obligations, and financial connections. As a result, the financial aspects of how disruptions propagate and recover feel underdeveloped in the simulation. It's more accurate to view the digital twin as a useful simulation approximation rather than an active corporate tool linked to live company systems. The machine learning component is fairly straightforward, relying on basic tabular Q-learning. While this keeps the approach easy to understand and quick to execute, it cannot effectively manage the complex, unpredictable scenarios that arise in actual logistics challenges. Transitioning to more advanced reinforcement learning techniques would be necessary to navigate multi-layered decisions with fluctuating demand and changing routes. When extending these ideas to other applications, the combination of real and simulated data may not seamlessly integrate into true corporate supply chains due to differences in scale, operating procedures, and data quality. The findings indicate that the basic method functions, but they do not demonstrate how effectively it would operate when implemented in a factory setting.

Internal validity is compromised by label leakage in the setup of artificial vulnerabilities. Because the vulnerability labels derive directly from variables that also serve as predictive features, the models may simply be reverse-engineering the labeling rules instead of uncovering genuine underlying patterns. This can inflate performance metrics, so a critical perspective should be applied to the results. External validity is limited because the manufactured network structure cannot replicate the complexities of a real global supply chain. The exclusion of actual purchasing habits, legal dependencies, and geopolitical factors means that these insights may not translate well outside a controlled environment. Construct validity suffers from the use of simulated risk measures and proxy emissions. Although these approximations aim to mimic real-world situations, they fail to reflect the actual mechanisms at play, particularly concerning sustainability issues and governance metrics. Statistical validity is also hampered by the lack of repeated test runs. Without extensive cross-validation or formal significance testing, the reported success rates should be treated as tentative estimates rather than definitive statistical conclusions.

7. Future Work

Improving this research later on will help make it more grounded and useful in real-world applications. A good starting point is to incorporate actual supplier data, such as genuine purchasing histories, shipping contracts, and multi-tier supplier relationships. Replacing the fictitious elements with real supply chain structures would enhance external validity and make the information easier to understand. Another effective approach is to advance the learning framework beyond basic Q-learning. Integrating deep reinforcement learning techniques like Proximal Policy Optimization, Deep Q-Networks, or Advantage Actor-Critic methods would enable the system to handle complex decisions, unpredictable demand changes, and continuous routing. This would allow for modeling how logistics strategies adapt during chaotic situations. Utilizing temporal graph structures, such as Temporal Graph Networks or Temporal Graph Attention Networks, is another viable upgrade. These models would help the system observe how supply chain connections evolve, how disruptions unfold day by day, and how risks flow through the network, rather than assuming the graph remains static.

Future versions should also consider text-based risk identification. Incorporating external data sources like news wires, trade sanction lists, port backlog updates, and public company reports would enable the system to transition from a passive simulation to an active risk monitor. This would provide managers with earlier warnings when a supply chain begins to face challenges. Directly linking the entire framework to active corporate software such as ERP, TMS, or WMS systems would significantly enhance its utility for daily operations. Connecting to live logistics data feeds would offer real-time decision-making support, finally bridging the gap between laboratory testing and actual warehouse implementation.

Additionally, making these graph models more explainable is a critical task for future iterations. Implementing tools like GNNExplainer, integrated gradients for graphs, or counterfactual reasoning would illuminate why the system identifies a vulnerability or selects a specific route. This transparency is essential for maintaining regulatory compliance and building trust with board members. Finally, future versions need to eliminate proxy emissions and inaccurate scores, replacing them with concrete sustainability metrics derived from carrier tracking and verified disclosures. This change would shift the sustainability module from a rough estimation process to a precise, auditable system of carbon tracking, allowing the framework to support daily shipping decisions and official environmental reporting simultaneously.

8. Conclusion

This study developed an integrated AI framework for modeling resilient and sustainable supply chain networks in a U.S. retail context by combining empirical demand data with transparently documented synthetic network augmentation. Using the M5 Forecasting dataset as the demand foundation, the framework linked forecasting, graph-based vulnerability modeling, digital-twin-inspired disruption simulation, Scope 3 emissions estimation, and reinforcement-learning-based rerouting optimization. The results show that network-aware methods provide a stronger basis for resilience analysis than forecasting alone. GraphSAGE performed best in the synthetic experimental setting, and the simulation results highlighted clear trade-offs between service recovery, cost, and emissions. These findings suggest that resilience and sustainability should be treated as joint optimization objectives.

A key contribution of this work is the explicit separation of empirical observations from synthetic assumptions. While the framework is not a production-ready solution, it provides a reproducible platform for testing integrated resilience and sustainability strategies. Future work should incorporate real supplier data, audited emissions indicators, and more advanced learning methods to improve external validity and practical deployment.

References

1. Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., … Pascanu, R. (2018). Relational inductive biases, deep learning, and graph networks. arXiv preprint arXiv:1806.01261.
2. Box, G. E. P., & Jenkins, G. M. (1970). Time series analysis: Forecasting and control. Holden-Day.
3. Carter, C. R., & Rogers, D. S. (2008). A framework of sustainable supply chain management: Moving toward new theory. International Journal of Physical Distribution & Logistics Management, 38(5), 360–387.
4. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794).
5. Chopra, S., & Sodhi, M. S. (2004). Managing risk to avoid supply-chain breakdown. MIT Sloan Management Review, 46(1), 53–61.
6. Choi, T. M., Wallace, S. W., & Wang, Y. (2018). Big data analytics in operations management. Production and Operations Management, 27(10), 1868–1883.
7. Christopher, M., & Peck, H. (2004). Building the resilient supply chain. The International Journal of Logistics Management, 15(2), 1–14.
8. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.
9. Hamilton, W., Ying, R., & Leskovec, J. (2017). Inductive representation learning on large graphs. In Advances in Neural Information Processing Systems (NeurIPS).
10. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780.
11. Hyndman, R. J., & Athanasopoulos, G. (2018). Forecasting: Principles and practice (2nd ed.). OTexts.
12. Ivanov, D. (2020). Predicting the impacts of epidemic outbreaks on global supply chains. International Journal of Production Research, 58(10).

13. Ivanov, D. (2021). Supply chain viability and the COVID-19 pandemic. International Journal of Production Research, 59(12), 3535–3552.
14. Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In International Conference on Learning Representations (ICLR).
15. Kritzinger, W., Karner, M., Traar, G., Henjes, J., & Sihn, W. (2018). Digital twin in manufacturing: A categorical literature review. IFAC-PapersOnLine, 51(11), 1016–1022.
16. Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2022). The M5 accuracy competition: Results, findings, and conclusions. International Journal of Forecasting, 38(4), 1346–1364.
17. McKinnon, A. (2018). Decarbonizing logistics: Distributing goods in a low carbon world. Kogan Page.
18. Seuring, S., & Müller, M. (2008). From a literature review to a conceptual framework for sustainable supply chain management. Journal of Cleaner Production, 16(15), 1699–1710.
19. Sheffi, Y. (2005). The resilient enterprise: Overcoming vulnerability for competitive advantage. MIT Press.
20. Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction (2nd ed.). MIT Press.
21. Tang, C. S. (2006). Perspectives in supply chain risk management. International Journal of Production Economics, 103(2), 451–488.
22. Tao, F., Zhang, H., Liu, A., & Nee, A. Y. C. (2019). Digital twin in industry: State-of-the-art. IEEE Transactions on Industrial Informatics, 15(4), 2405–2415.
23. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2018). Graph attention networks. In International Conference on Learning Representations (ICLR).
24. Watkins, C. J. C. H., & Dayan, P. (1992). Q-learning. Machine Learning, 8, 279–292.
25. Wiedmann, T. O., Schandl, H., Lenzen, M., Moran, D., Suh, S., West, J., & Kanemoto, K. (2015). The material footprint of nations. Proceedings of the National Academy of Sciences, 112(20), 6271–6276.
26. Wiedmann, T., & Minx, J. (2008). A definition of carbon footprint. In Ecological Economics Research Trends (pp. 1–11).