Nexus Risk Platform

> **Supply Chain Resilience Predictor**
> *AI-powered disruption prediction for global semiconductor supply chains.*

Live Demo (https://nexus-risk-platform-vedq.vercel.app)
API Documentation (https://nexus-risk-platform-production.up.railway.app/docs)
License MIT (LICENSE)

Overview

The **Nexus Risk Platform** is an advanced analytics dashboard designed to monitor, analyze, and predict supply chain disruptions in the critical Taiwan-to-US West Coast semiconductor corridor.

By integrating real-time vessel tracking, weather data, and geopolitical news with Graph Neural Networks (GNN) and Explainable AI (XAI), Nexus provides logistics managers with actionable insights to mitigate risks before they impact production.

Key Features

- **Real-time Tracking**: Live monitoring of key shipping routes and vessel positions.
- **Risk Prediction**: AI-driven delay probabilities based on multi-factor analysis.
- **Network Analysis**: Graph Neural Network (GNN) modeling of port cascading failures.
- **Explainable Insights**: "Why" behind the risk—weather, congestion, or geopolitical events.
- **Interactive Dashboard**: Beautiful, responsive UI built with Next.js and Mapbox/Leaflet.

Architecture

The platform follows a modern microservices-inspired architecture:

```
graph TD
    subgraph Frontend["Frontend (Vercel)"]
        UI[Next.js Dashboard]
        Viz[Recharts & Leaflet Maps]
        State[React Query / Zustand]
    end

    subgraph Backend["Backend (Railway)"]
        API[FastAPI Server]
        Auth[JWT Authentication]

        subgraph Services
            LDS[Live Data Service]
            GNN[Graph Neural Network]
            XAI[Explainable AI]
            Pred[Prediction Engine]
        end

        subgraph Workers
            News[News Collector]
            Weather[Weather Collector]
            Vessel[Vessel Collector]
        end
    end

    subgraph Data["Data Layer (Supabase)"]
        DB[(PostgreSQL)]
        Raw[Raw Logs]
    end

    subgraph External["External APIs"]
        MT[MarineTraffic]
        OW[OpenWeather]
        NewsAPI[NewsAPI]
    end

    UI --> API
    API --> Services
    Services --> DB
    Workers --> External
    Workers --> DB

    classDef box fill:#f9f9f9,stroke:#333,stroke-width:2px;
    class UI,API,DB box
```

Tech Stack

Frontend

- **Framework**: Next.js 14 (App Router)
- **Language**: TypeScript

- **Styling**: Tailwind CSS, Framer Motion
- **Visualization**: Recharts, Leaflet (React-Leaflet)
- **State Management**: React Query

Backend

- **Framework**: FastAPI (Python 3.10+)
- **ORM**: SQLAlchemy (Async)
- **ML Frameworks**: PyTorch, PyTorch Geometric, Scikit-learn
- **Data Processing**: NumPy, Pandas, NetworkX

Infrastructure

- **Database**: PostgreSQL (via Supabase)
- **Hosting**: Railway (Backend), Vercel (Frontend)
- **CI/CD**: GitHub Actions (Linting, Testing, Data Collection)

Quick Start

Prerequisites

- Python 3.10+
- Node.js 18+
- PostgreSQL / Supabase account

1. Clone the Repository

```
git clone https://github.com/veerryait/nexus-risk-platform.git
cd nexus-risk-platform
```

2. Backend Setup

```
cd backend
python -m venv venv
source venv/bin/activate  # Windows: venv\Scripts\activate
pip install -r requirements.txt

# Create .env file (see .env.example)
cp .env.example .env

# Run Server
uvicorn app.main:app --reload
```

3. Frontend Setup

```
cd frontend
npm install

# Create .env.local file
echo "NEXT_PUBLIC_API_URL=http://localhost:8000" > .env.local

# Run Dev Server
npm run dev
```

Visit http://localhost:3000 to view the dashboard.

Documentation

Detailed documentation is available in the /docs directory:

- **Technical Documentation** (docs/TECHNICAL_DOCS.md): Deep dive into data pipelines, ML models, and API endpoints.
- **User Guide** (docs/USER_GUIDE.md): How to interpret risk scores and use the dashboard.

Contributing

1. Fork the Project
2. Create your Feature Branch (git checkout -b feature/AmazingFeature)
3. Commit your Changes (git commit -m 'Add some AmazingFeature')
4. Push to the Branch (git push origin feature/AmazingFeature)
5. Open a Pull Request

License

Distributed under the MIT License. See LICENSE for more information.

Technical Documentation

1. System Architecture

Nexus Risk Platform is built on a modular, containerized architecture designed for scalability and maintainability.

1.1 High-Level Overview

- **Frontend**: Next.js application served via Vercel Edge Network.
- **Backend**: FastAPI service running in Docker on Railway.
- **Database**: PostgreSQL managed by Supabase.
- **Automation**: GitHub Actions for scheduled data ingestion.

---

2. Data Pipeline

The data pipeline aggregates heterogeneous data sources into a unified risk model.

2.1 Data Sources

Domain	Source	Frequency	Purpose
Vessels	MarineTraffic / AIS	4 Hours	Live position, speed, heading
Weather	OpenWeatherMap	12 Hours	Typhoon tracking, wave height
News	NewsAPI / GDELT	6 Hours	Port strikes, trade policy changes
Ports	Internal DB	Static	Port coordinates, capacity (TEUs)

2.2 Ingestion Flow

1. **Collectors**: Specialized Python scripts (backend/collectors/) run via GitHub Actions.
2. **normalization**: Raw JSON/XML is parsed and normalized into Pydantic models.
3. **Storage**: Cleaned data is upserted into Supabase tables (vessel_positions, weather_alerts).
4. **Signal Generation**: live_data_service.py aggregates recent data to compute Operational Risk.

---

3. Machine Learning Models

Nexus uses a hybrid approach combining rule-based heuristics and Graph Neural Networks.

3.1 Random Forest Regressor (Route Risk)

- **Goal**: Predict delay (in days) for individual routes.
- **Features**:
  - distance_remaining (Normalized)
  - average_speed (vs design speed)
  - weather_severity_index (0-1)
  - port_congestion_level (0-1)
- **Implementation**: Scikit-learn (app/ml/predict.py).

3.2 Graph Neural Network (Network Risk)

- **Goal**: Identify cascading risks in the global port network.
- **Architecture**:
  - **Nodes**: Ports (Features: Capacity, Congestion, Weather)
  - **Edges**: Shipping Routes (Features: Distance, Transit Time)
  - **Model**: Graph Attention Network (GAT) or GCN.
- **Inference**:
  - **GPU Mode**: Uses PyTorch Geometric if CUDA/MPS is available.
  - **Standard Mode**: Falls back to specialized NumPy matrix operations for CPU environments (like Railway).
- **Output**: Network Risk Score (0-100) for the entire system and individual nodes.

3.3 Explainable AI (XAI)

- **Mechanism**: Uses Integrated Gradients (conceptually) to attribute risk scores to specific feature inputs.
- **Output**: Natural language explanations (e.g., "High risk due to Typhoon near Kaohsiung").

4. API Reference

The backend exposes a RESTful API via FastAPI.

Base URL: /api/v1

Key Endpoints

Routes

- GET /routes/: List active shipping routes with current status.
- GET /routes/{id}: Detailed telemetry for a specific route.

Predictions

- POST /predict/: Get risk assessment for a hypothetical or real route.

```
{
  "route_id": "TW-US-001",
  "specific_features": { "weather_impact": 0.8 }
}
```

GNN

- GET /gnn/predict: Get structural network risk.
- GET /gnn/network: Get graph topology for visualization.

5. Development & Deployment

5.1 Local Development

The project uses dotenv for configuration. API keys must be set in .env.

5.2 Deployment Strategy

- **Backend**: Dockerfile builds a lightweight Python image. Deployed to Railway from the main branch.
- **Frontend**: Next.js build pipeline on Vercel. Automagically deploys on push.
- **Database**: Migrations managed by SQLAlchemy (currently using Base.metadata.create_all).

5.3 Monitoring

- **Logs**: Application logs are routed to stdout (captured by Railway).
- **Health Check**: /health endpoint serves as a heartbeat for uptime monitors.

User Guide

Welcome to the Nexus Risk Platform! This guide will help you navigate the dashboard and understand the supply chain risk insights provided by our AI models.

1. The Dashboard Interface

The dashboard is your command center for monitoring the Taiwan-US semiconductor supply chain.

1.1 Map View

The central component is the interactive map.

- **Blue Lines**: Active shipping routes.
- **Orange/Red Dots**: Ports, colored by congestion level.
- **Ship Icons**: Real-time position of container vessels.
    - *Click on a ship* to see details like Speed, Heading, and Destination.

1.2 Quick Stats ⚡

Located at the top of the dashboard:

- **Total Routes**: Number of active monitored lanes.
- **Avg On-Time %**: Network-wide reliability metric.
- **High Risk Routes**: Count of routes currently flagged with critical delays.

1.3 AI Risk Analyst

On the right sidebar, you'll find the **AI Summary**.

- **What it does**: Our XAI (Explainable AI) engine reads raw data (weather, news, satellite) and summarizes it into plain English.
- **Example**: *"High risk detected due to Typhoon In-Fa approaching Taipei Port. Expect 48h delays."*

---

2. Interpreting Risk Scores

Nexus uses a dual-layer risk scoring system. It is important to understand the difference between **Route Risk** and **Network Risk**.

2.1 Route Risk (Operational)

Found in the "Active Routes" table.

- **Focus**: A specific ship on a specific path.
- **Factors**: Current speed, local weather, engine status.
- **Levels**:
    - **Low**: On schedule.
    - **Medium**: Minor delays likely (1-2 days).
    - **High**: Critical disruption (3+ days delay).

2.2 Network Risk (Structural)

Found in the "Graph Neural Network Analysis" section.

- **Focus**: The entire supply chain system.
- **Factors**: Global port congestion, interconnected dependencies.
- **Why it matters**: Even if *your* ship is fine, if the destination port (e.g., Los Angeles) is gridlocked, your **Network Risk** will be high (50%+).
- **Visualization**: The "Network Risk" nodes show which ports are the bottlenecks in the system.

3. Common Questions (FAQ)

Q: Why is "Route Risk" Low but "Network Risk" is High?
A: This is common! Your ship might be sailing perfectly (Low Operational Risk), but heading towards a port that is completely overwhelmed (High Structural Risk). Nexus separates these so you can distinguish between "my ship is broken" vs "the system is broken."

Q: How often is data updated?
A:

- **Vessel Positions**: Every 4 hours.
- **Weather**: Every 12 hours.
- **News**: Every 6 hours.

Q: Can I simulate a disruption?
A: Yes! Use the "Predict" API endpoint (see technical docs) to ask "What if weather impact increases to 90%?"

4. Support

For technical support or to report a bug, please open an issue in our GitHub Repository (https://github.com/veerryait/nexus-risk-platform/issues).