Graph Neural Network-Based Demand Forecasting

Lena Schill
Student ID: 12404231

January 2026

Abstract

This report shows the development and results of a deep learning model for multivariate demand forecasting, combining Graph Neural Networks with Long Short-Term Memory networks. The goal of the project was to tackle the problem of inter-product dependencies by modeling co-purchase patterns as a static graph, whose embeddings were fused with temporal inputs. The report provides a rationale for the architectural choices, the inference mechanism, and key insights into the challenges encountered in this project.

1 Introduction and Problem Statement

Most traditional demand forecasting models in retail primarily rely on univariate time series analysis (e.g., ARIMA or simple LSTMs), treating each product's sales history independently. This methodology is often not enough for capturing the reality of consumer behavior, where purchases are highly correlated (e.g., promotional effects, substitution, or complementary items). The inability to model these dependencies might result in underestimation of complex market effects and poor performance, particularly for products characterized by low volume or high demand volatility. Thus the core challenge of this project was to tackle this problem by developing a model capable of leveraging two distinct data modalities, including the structure of product relationships and the sequential patterns of time series data, to produce context-aware demand forecasts across the entire product catalog.

This requires a deep learning approach because traditional models are mathematically limited to tabular or linear structures, making them unable to read the complex relationships between products. Deep learning, however, facilitates Representation Learning, where the model doesn't just look at numbers, but creates a high-dimensional embedding space. In this space, complementary items (like pasta and sauce) are positioned close together. By using architectures like Graph Neural Networks, we can inject the structure of the retail catalog directly into the math. The model then uses a fusion layer to combine these relational insights with temporal patterns from LSTMs or Transformers. This allows the system to understand that a promotion on Product A will likely cannibalize the demand for Product

B. This is a level of context-awareness that an independent statistical model simply cannot achieve.

2 Methodology

In the following sections We will explain the architectural choices, the chosen loss function and the inference strategy.

2.1 Architecture

The selection of the GNN-LSTM hybrid architecture was driven by the specialized strengths of each component in handling the data science challenges of retail forecasting. The primary necessity for a deep learning approach, as established in the introduction, arises from the requirement to move beyond univariate bottlenecks where products are treated as isolated entities. While traditional statistical models or standard machine learning algorithms rely on flattened, tabular data, they struggle to ingest the complex, non-linear dependencies found in retail ecosystems. To solve this, the GNN-LSTM architecture was developed to perform multimodal representation learning, allowing the model to understand both the timing of sales and the structural relationships between items.

2.1.1 Graph Neural Network

The first component, the Graph Neural Network, specifically addresses the challenge of modeling the product catalog as a non-Euclidean structure. Unlike standard tabular data or images where observations follow a rigid, grid-like format (Euclidean space), a retail network is defined by irregular relationships. For example, a bottle of red wine might have strong edges connecting it to corkscrews and high-end cheeses, but no relationship to laundry detergent. In this non-Euclidean space, products have varying degrees of connectivity that cannot be captured by fixed-size filters or simple distance metrics. Also standard deep learning techniques like MLPs fail to aggregate this information across these connections effectively. However by utilizing GNN layers the model performs message passing between nodes. This means a product's feature representation is dynamically informed by the demand states of its neighbors, so essentially if charcoal sales spike, the GNN propagates this signal to the "Barbecue Grill" node. This leads to a rich relational embedding that captures market effects like complementarity more effectively than traditional one-hot encoding.

2.1.2 Long Short-Term Memory

Complementing this relational insight, the Long Short-Term Memory network serves as the temporal engine of the architecture. While the GNN maps the "where" in the product network, the LSTM models the "when." Time series data in retail is often plagued by non-stationarity and annual seasonality. For instance, the demand for sunblock follows a clear yearly cycle with high sensitivity to short-term weather noise. To process the multimodal input, the LSTM receives a concatenated vector at each timestep with the raw temporal features (like price or previous day's sales) merged with the static relational embedding from

the GNN. The LSTM's power then lies in its gating mechanism, which acts as a information filter. The forget gate decides which past information (such as a temporary sales spike caused by a viral TikTok trend of people eating pudding with a fork) is no longer relevant to the future trend. The input gate then determines which new information from the GNN-informed feature vector (e.g., a new competitor price change) should be stored in the cell state, which acts as the model's long-term memory. Finally, the output gate decides which parts of that cell state will be used to generate the hidden state for the current forecast.

This sequential processing ensures that the model recognizes temporal dynamics that simple regressive models would overlook. For example, the LSTM can remember that demand usually rises three weeks before a major holiday and forget noise of a random event. The core innovation of this project lies in this deep feature fusion. By injecting the GNN's relational context into the LSTM at every step, the model's hidden state evolution is continuously conditioned on the product's position within the wider market graph. This allows the system to generalize patterns from high-volume products to those with high demand volatility or sparse history, directly addressing the limitations of independent univariate forecasting.

2.2 Loss Function

In retail demand forecasting, selecting the appropriate loss function is as critical as the architecture itself, as the data is discrete, non-negative, and highly skewed. We evaluated three primary loss functions, progressing from standard regression to complex probabilistic mixture models, ultimately selecting the one that provided the best balance between statistical accuracy and training stability.

As a baseline, Mean Squared Error was tested. Mathematically, minimizing MSE is equivalent to maximizing the likelihood of a Gaussian distribution. This proved inappropriate for the Instacart dataset because it treats the high frequency of zero-sales periods as outliers to be averaged out, which drives the mean prediction toward zero and fails to capture critical demand spikes. Furthermore, MSE lacks the constraints necessary to prevent negative demand predictions, which are physically impossible in a retail environment.

To address these flaws, we initially experimented with the Zero-Inflated Negative Binomial loss. Theoretically, ZINB is the most sophisticated choice for sparse retail data because it uses a mixture model to handle structural zeros while allowing the variance to grow independently of the mean through a dispersion parameter. However, during experimental trials, the ZINB approach introduced significant gradient instability. Because the model was forced to simultaneously optimize three separate parameters, zero-probability, dispersion, and the mean, the loss landscape became overly complex. Given the hardware limitations of the project, this complexity led to poor convergence and inferior performance compared to simpler models.

The final model utilizes Poisson Negative Log-Likelihood, which assumes the probability of demand follows a Poisson distribution. This was found to be the optimal choice for this project as it offered superior numerical stability. Unlike ZINB, the Poisson distribution relies on a single parameter, making it significantly easier for the optimizer to converge under limited computational power. It naturally prevents negative predictions and is mathematically designed for integer count data. Despite the theoretical risk of overdispersion, where the mean is forced to equal the variance, Poisson NLL empirically produced the lowest RMSE

on the test set. By applying a ReLU activation to the final output layer to ensure a positive rate parameter, we achieved a robust forecasting system that outperformed more complex alternatives in a resource-constrained environment

2.3 Inference

To produce forecasts beyond observed history, the model transitions from single-step training to multi-step inference. The forecast horizon decision was addressed using an Iterative Multi-Step strategy with $H = 1$, where the model is trained to predict only the next time step ($\hat{y}_{t+1}$). This choice improves training stability and data efficiency but requires recursive forecasting at inference time.

For a $T_{forecast}$ horizon, predictions are generated sequentially, where each output is appended to the input window, the oldest observation is dropped, and the updated sequence is fed back into the GNN-LSTM. This rolling-window recursion treats prior predictions as pseudo-ground truth.

Alternative approaches were evaluated but rejected due to computational constraints. Direct multi-step prediction requires horizon-specific output weights, causing linear growth in parameters that exceeded GPU memory limits. Seq2Seq architectures introduce a separate decoder, doubling training cost and increasing inference latency beyond available compute capacity.

The recursive strategy however keeps the model lightweight by learning only single-step dynamics while preserving relational reasoning through repeated GNN embeddings. The main trade-off is added implementation complexity in managing high-dimensional recursive tensor updates across graph-connected product histories.

3 Main Take Aways

The primary takeaway from this project is that while graph-based demand forecasting is conceptually appealing, its practical effectiveness is highly constrained by data quality, sparsity, and system complexity. Although the proposed GNN–LSTM architecture was theoretically capable of modeling inter-product dependencies, the empirical results indicate that these benefits did not translate into improved forecasting accuracy under the given conditions.

A central challenge was extreme data sparsity, which severely limited the model's ability to learn meaningful temporal and relational patterns. In a catalog with roughly 50,000 products, the vast majority of time steps contain zero demand. This produced a vanishing signal problem in the LSTM and undermined the learning of stable graph embeddings, as message passing propagated mostly zero-valued information across the network. As a result, the relational context encoded by the GNN often lacked sufficient signal to improve downstream forecasts. Closely related to this was the misalignment between dataset design and forecasting objectives. The Instacart dataset is fundamentally optimized for recommender systems, where user–item interactions are modeled at a transactional level rather than as continuous time series. Converting this data into equally spaced demand sequences and a static co-purchase graph required extensive preprocessing and aggregation. Despite this effort, the resulting signals remained noisy and sparse, limiting the effectiveness of both the

GNN and the temporal model.

From an optimization perspective, the project revealed that statistically expressive loss functions can become liabilities in low-signal regimes. Although zero-inflated models are theoretically well suited for retail demand, the Zero-Inflated Negative Binomial loss proved unstable in practice. The scarcity of non-zero observations provided insufficient gradient information to reliably learn multiple distributional parameters, leading to poor convergence. The simpler Poisson NLL improved numerical stability but did not resolve the underlying information deficit in the data. The project also highlighted significant engineering and scalability challenges. Managing high-dimensional tensors across tens of thousands of products, fusing static graph embeddings with dynamic sequences, and implementing recursive multi-step inference introduced substantial implementation complexity. These challenges increased the risk of subtle bugs and constrained experimentation, without yielding proportional gains in model performance.

Overall, the results suggest that architectural sophistication alone is insufficient to overcome fundamental data limitations. Graph-based deep learning for demand forecasting requires not only relational structure, but also dense, informative temporal signals and datasets explicitly designed for forecasting tasks. In the absence of these conditions, simpler models with fewer assumptions may offer more reliable and interpretable performance.

4 Conclusion

If I were to start the project again, the primary focus would shift from architectural sophistication to data suitability and quality. While the GNN–LSTM framework is theoretically well suited for modeling inter-product dependencies, this project demonstrated that such complexity cannot compensate for extreme sparsity and domain misalignment. Over 120 hours were spent on this project, with a significant portion devoted to data preprocessing and the fusion of multiple data modalities, efforts that were initially underestimated. Selecting a dataset explicitly designed for demand forecasting, with denser temporal signals and clearer product-level dynamics, would be a prerequisite before introducing graph-based relational modeling.

From a modeling perspective, future iterations would prioritize simpler baselines and stronger empirical validation before scaling to complex multimodal architectures. Establishing robust univariate or multivariate benchmarks would provide a clearer reference point for evaluating the marginal value of graph embeddings. Additionally, experimenting with dynamic or learned product relationships, rather than a static co-purchase graph, may better reflect evolving consumer behavior and reduce reliance on noisy relational signals.

Finally, this project underscores the importance of balancing theoretical expressiveness with practical constraints. High-dimensional tensor management, recursive inference, and advanced probabilistic losses significantly increased implementation complexity while offering limited performance gains. A more iterative development strategy grounded in data exploration, preprocessing, loss stability, and computational feasibility would likely yield more reliable and interpretable forecasting outcomes in large-scale retail settings.