Applied Deep Learning

GNN-based Demand Forecasting

Lena Schill: 12404231
14.01.2025

Traditional demand forecasting ignores inter-product dependencies, leading to inaccurate predictions.

Problem

Traditional retail demand forecasting treats each **product independently, failung to capture correlations.**

This independence leads to **poor performance** for low-volume or highly volatile products and an **underestimation of complex market dynamics.**

Existing statistical and **shallow ML models can't jointly model temporal demand patterns** and **structured relationships** between products.

Solution

Model the **retail catalog as a graph** and use a GNN to leanr relational embeddings that caputure product-to-product dependencies.

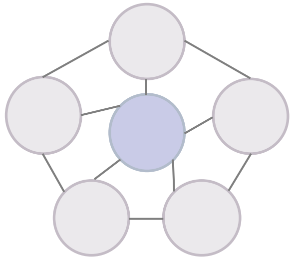
Combine these **relational embeddings** with **sequential sales** features using an LSTM to jointly model the "where" (product relationships) and "when" (temporal dynamics).

Fuse graph-based context and time-series learning into a **single deep learning architecture** to produce context-aware demand forecasts across the full product catalog.

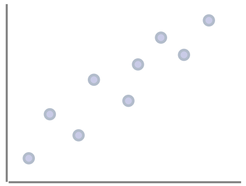
The model combines GNN-learned product relationships with LSTM temporal patterns to generate context-aware, multi-step demand forecasts

Input Modalities

Product Graph



Time Series Data



Relational Modeling

- Products as nodes, edges capture complement/substitute relationships
- Propagates signals to learn cross-product embeddings
- **Example:** Pasta promo → boosts sauce, reduces rice → guides forecasts

Feature Fusion
GNN embeddings are concatenated with per-timestep time series features

Temporal Modeling

- Products as nodes, edges Historical sales per product
- Learns trends, seasonality, and lagged effects from sequential data
- **Example:** Sauce demand rises before summer → LSTM captures seasonal spike in forecasts

Forecast

- LSTM outputs multi-step demand predictions per product
- **Recursion:** Each predicted step feeds into the next to generate future forecasts
- **Example:** Predicted pasta sales → inform next-day sauce demand → full week forecast updated

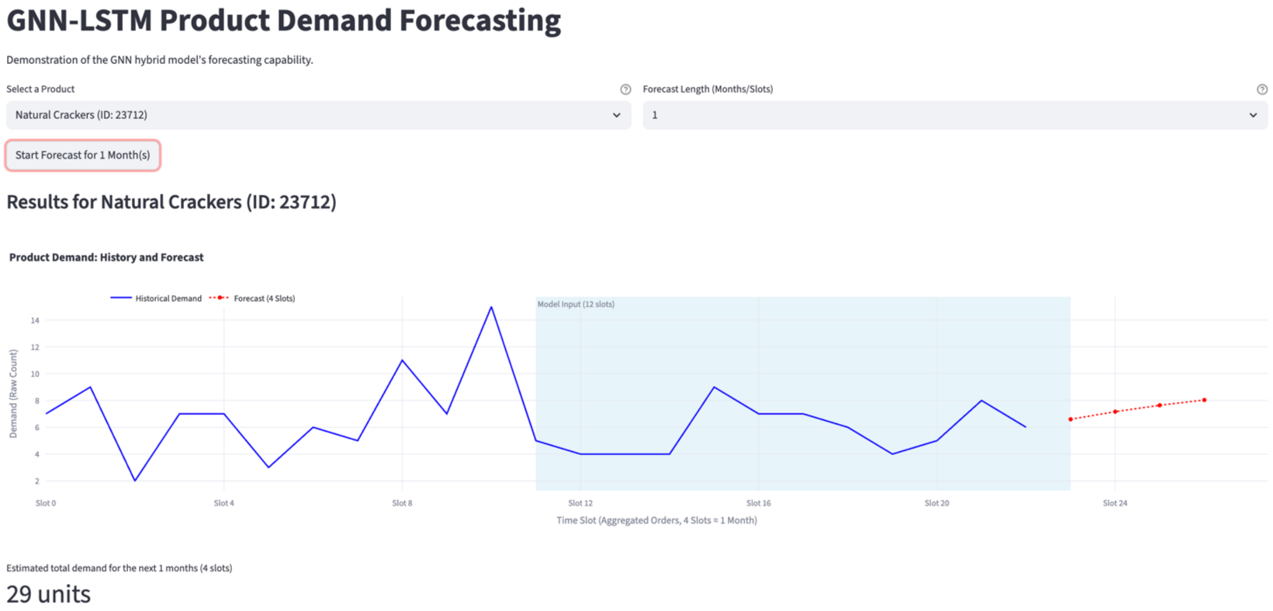
Results showed that the model first appeared promising with very low training Loss, but high test Loss revealed overfitting and limited generalization

Loss

Metric	Target	Achieved Value
Poisson Negative Log-Likelihood (NLL) Loss	Minimize loss value (target is task-dependent and converges to the minimum possible NLL)	**0.011159**

Metric	Target	Achieved Value
Root Mean Squared Error (RMSE) on Unscaled Demand	Achieve an RMSE value below a threshold relevant to business constraints (Target RMSE ≤ 2.5 units)	**2.913205**

Demo Application



- Training Poisson NLL is extremely low (0.011), but test RMSE is 2.9, showing limited generalization -> Overfitting
- Using RMSE on test data makes errors more interpretable in the original scale of the target.
- Model struggles on unseen data, likely memorizing training patterns rather than learning robust representations.

Learnings and Conclusion

Data is key

Sparse, noisy signals limited model learning

Complexity isn‘t enough

GNN–LSTM doesn’t always beat simpler baselines.

Loss Matters

Advanced losses are often unstable, Poisson NLL more stable but limited

Engineering overhead

High dimensional tensors and recursive inference added complexity

Future Focus

Prioritize data quality, simple baselines, and benchmarking before complex models.