UNIVERSITY OF MALAYA

WQF7023 Artificial Intelligence Research Project 2

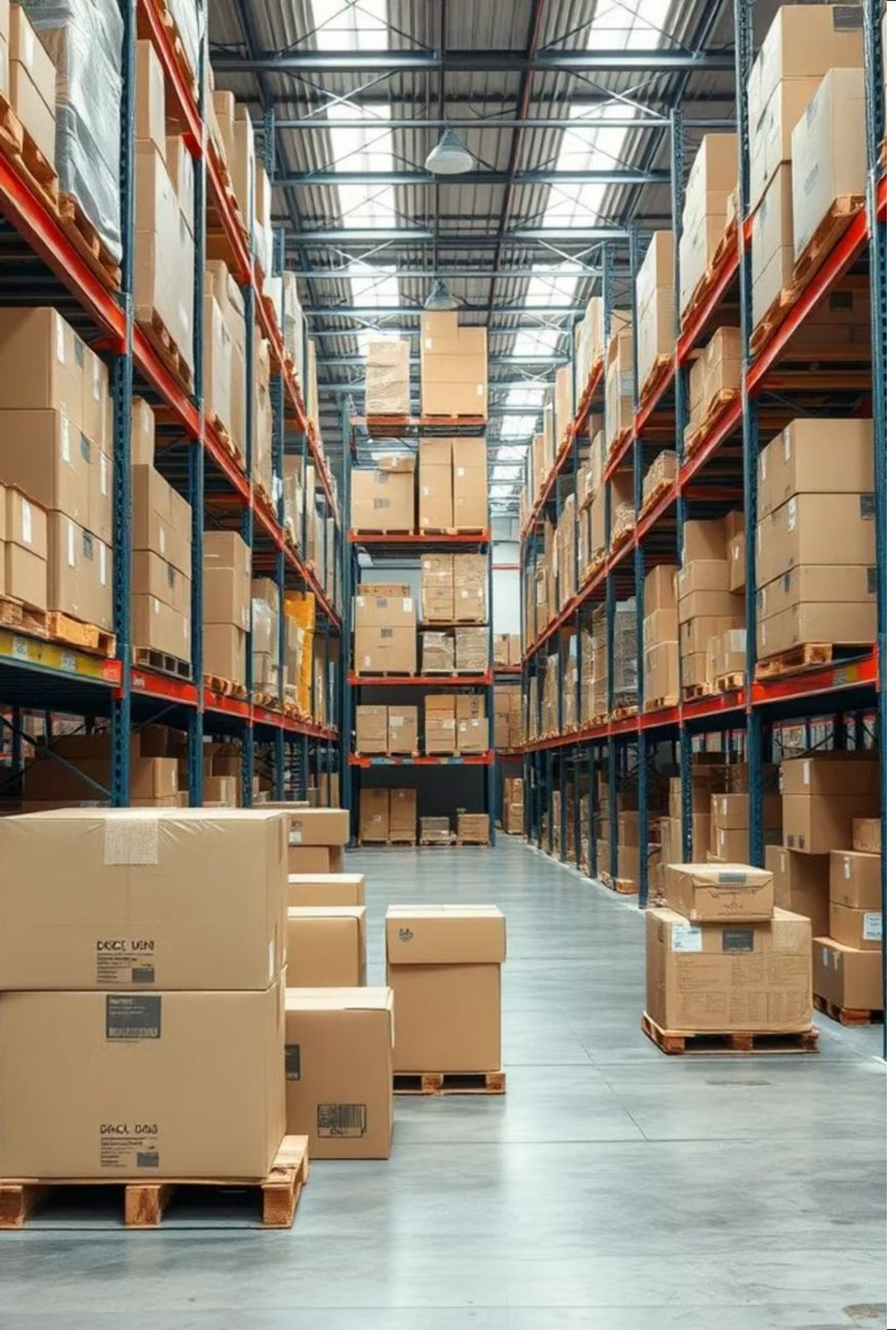
2025/2026 Semester 1

Explainable Demand Forecasting for Retailers using Graph Neural Networks and Temporal Fusion Transformers

Supervisor: ASSOCIATE PROF.TS.DR.Aznul Qalid Bin MD Sabri

Student: 23115588 Maksatbek kyzy Aizhan

Content of the Presentation



Introduction

Demand forecasting is central to **retail success**, enabling **efficient inventory and resource management**.

Complex supply chains and logistics increase the need for **advanced, AI-driven forecasting models**.

This research explores **GNN**, **TFT**, and **hybrid GNN-TFT models**, prioritizing both **predictive accuracy** and **model explainability**.

Explainable AI builds **trust and transparency**, ensuring retail stakeholders can understand and act on **model insights**.

Research Questions and Objectives

1

Compare GNN, TFT, and Hybrid Models for Retail Demand Forecasting Accuracy.

Which approach yields higher accuracy in retail demand forecasting: GNN, TFT, or a hybrid GNN-TFT model?

2

Identify Key Features Impacting Model Predictive Accuracy.

Which features within the retail dataset have the greatest and least impact on the predictive accuracy of these models?

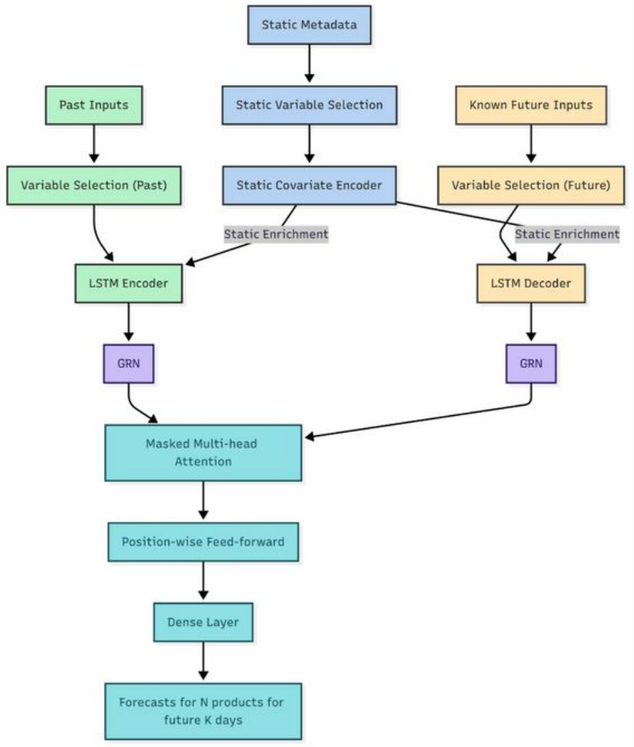
3

Determine Effective Architectural Improvements for Model Optimization.

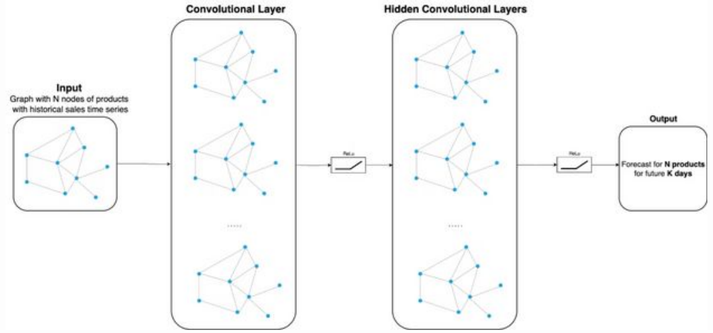
What architectural improvements are most effective for optimizing the performances of models?

Model Architectures

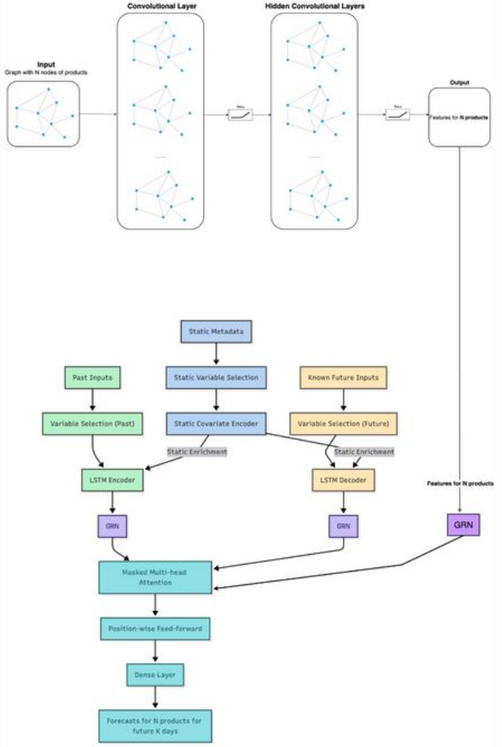
TFT



GNN



Hybrid GNN-TFT



Data - Favorita Grocery

33 product families

54 stores

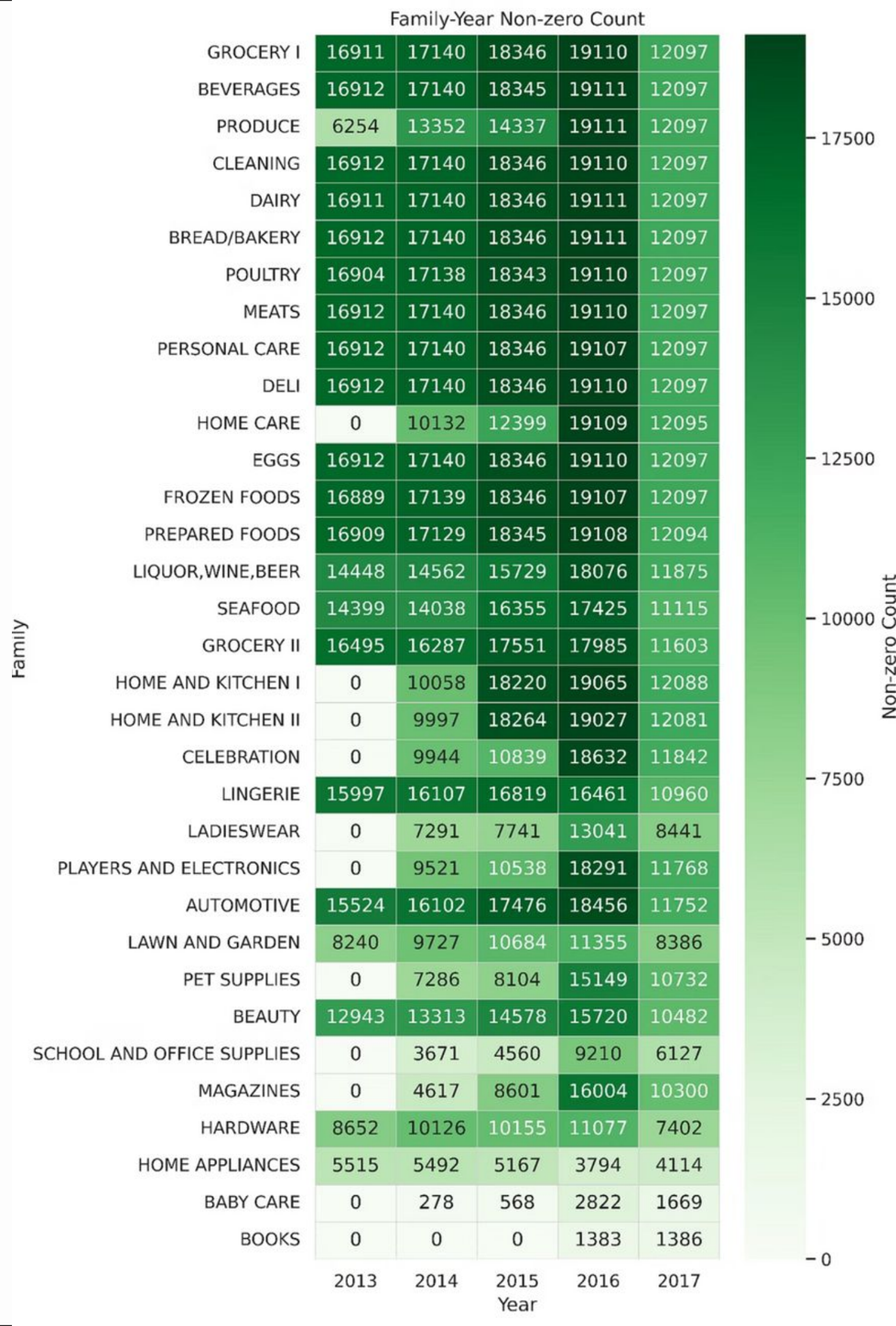
5 store types

17 store clusters

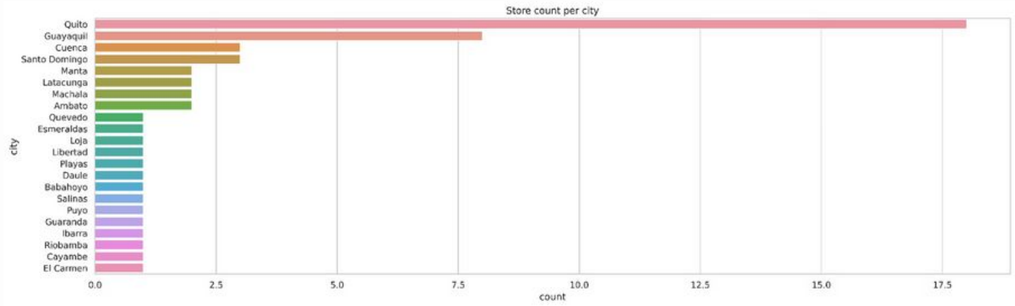
22 states

Grocery and household items showed high daily **non-zero sales** from 2013 to 2016, with a peak in 2016 and **a drop** in 2017.

Home care, baby care, and books either **started later or have much sparser sales**.

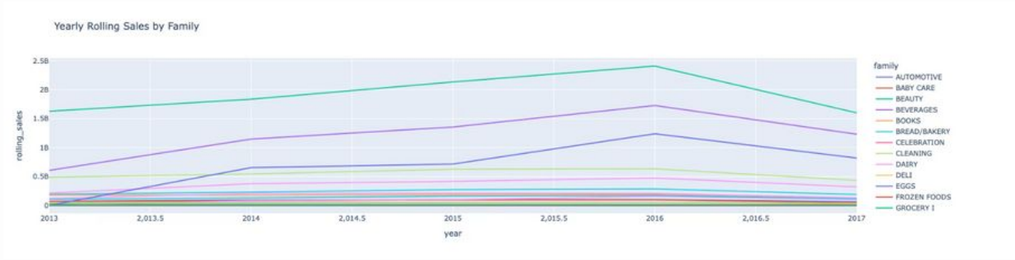


Analyzing Favorita Grocery Sales Trends



Store Distribution Across Cities

A significant number of Favorita stores are concentrated in Quito, one of 22 cities. This uneven distribution highlights notable sales imbalances across different urban areas within the dataset.



Temporal Sales Trends

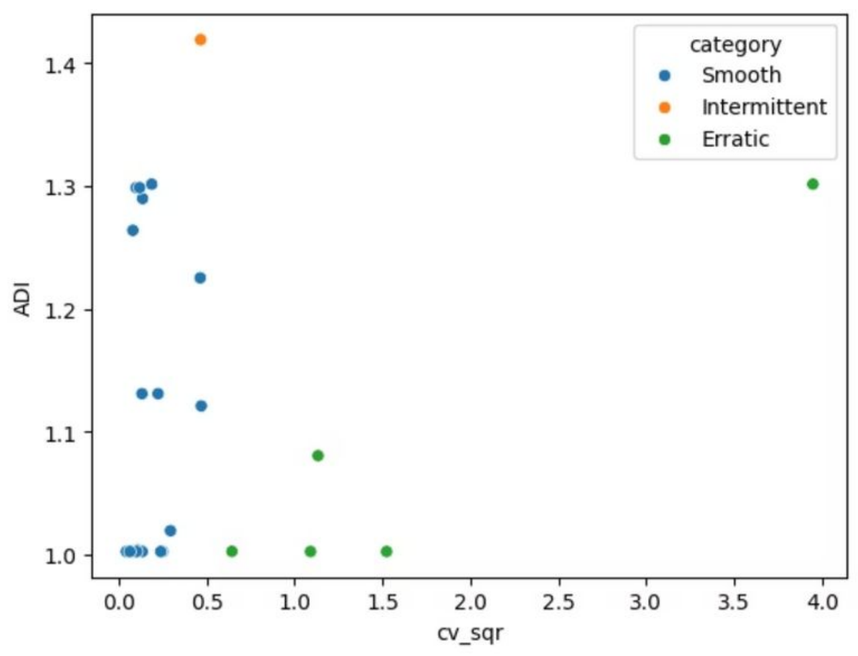
]Consistent seasonality and occasional spikes in sales when viewed at yearly and monthly frequencies. Overall, most product families demonstrate remarkable stability in their sales patterns.

Sales Performance by Product Family

This chart visualizes the amount of sales over a specific periodic duration for various product families, offering insights into their individual performance. Having the Grocery I in top sales quantity.

Forecastability

Measured using Average Demand Interval (ADI) and the squared Coefficient of Variation (CV^2).



Type	Range	Description
Smooth	ADI < 1.32 CV^2 < 0.49	Demand is regular both in timing and magnitude, leading to relatively straightforward forecasts with low expected error.
Intermittent	ADI ≥ 1.32 CV^2 < 0.49	Quantities show little variation, but demand intervals are irregular and often lengthy. Specialized methods are needed, and forecast errors tend to be higher.
Erratic	ADI < 1.32 CV^2 ≥ 0.49	Demand occurs at regular intervals, but quantities vary substantially. This makes forecast accuracy unstable and difficult to maintain.
Lumpy	ADI ≥ 1.32 CV^2 ≥ 0.49	

Prediction Model Scope: Category vs. Entire Sales

Developed at the product family (category) level (33 families), not individual SKUs.

Advantages:

- Captures robust sales patterns and seasonality due to reduced volatility at the category level.
- Yields more stable and accurate forecasts, especially when SKU-level data is sparse or highly variable.
- Reduces computing cost significantly compared to building models for each SKU.

Rationale:

- Effective as an initial step to understand broad sales trends and optimize resource allocation.
- Once established, the approach can be extended to SKU-level forecasting for detailed inventory planning.

Note:

SKU-level forecasting is ultimately preferred by store managers, and can be implemented as data and modeling processes mature.

Results: Forecasting Models Evaluated

Baseline Models

ARIMA and Linear Regression, providing foundational comparisons and initial performance benchmarks for sales forecasting.

Temporal Fusion Transformers (TFT)

An advanced neural network designed for robust time series forecasting, effectively capturing complex temporal dependencies.

Spatio-Temporal Graph Neural Networks (STGNN)

An innovative approach for modeling intricate spatial and temporal relationships within the sales data.

Hybrid GNN-TFT

Combines the unique strengths of Spatio-Temporal Graph Neural Networks and Temporal Fusion Transformers for enhanced predictive accuracy and robustness.

1. Baseline Models - ARIMA and Linear Regression

The Autoregressive Integrated Moving Average (ARIMA) model is a statistical method for analyzing and forecasting time-series data.

In this study, the ARIMA model was implemented with the parameter order (2,1,3). This configuration incorporates past values, applies differencing to remove trends, and leverages past forecast errors to predict future sales (see Section 5.2.1 for details).

ARIMA Performance Metrics:

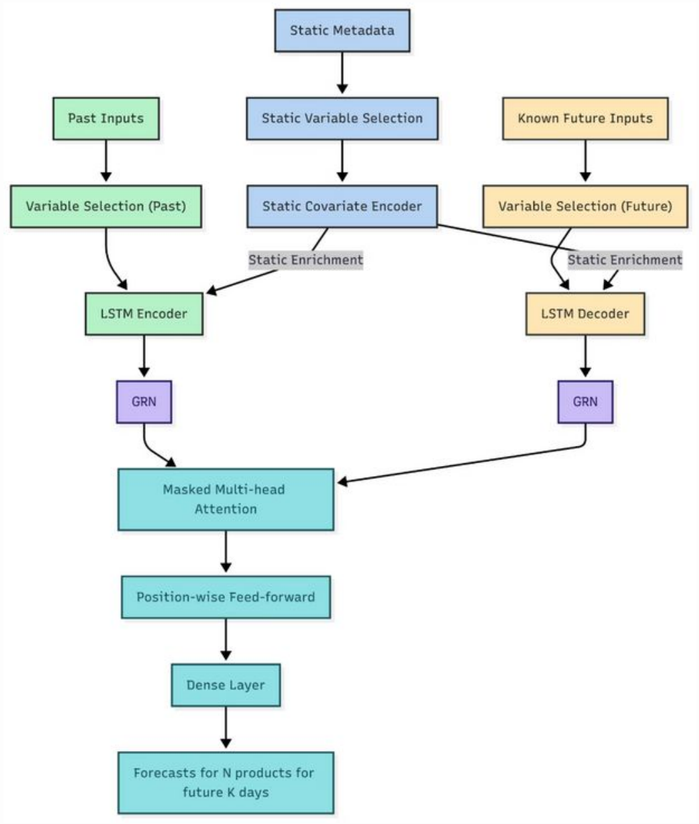
- MAE: 4175.60
- WAPE: 0.1630
- SMAPE: 0.3366

Basic Linear Regression (LR) model is included as a second baseline for comparison. LR is widely used due to its simplicity and its ability to model linear relationships between predictor variables and the target sales variable.

Linear Regression Performance Metrics:

- MAE: 315.74
- WAPE: 0.8446
- SMAPE: 1.4174

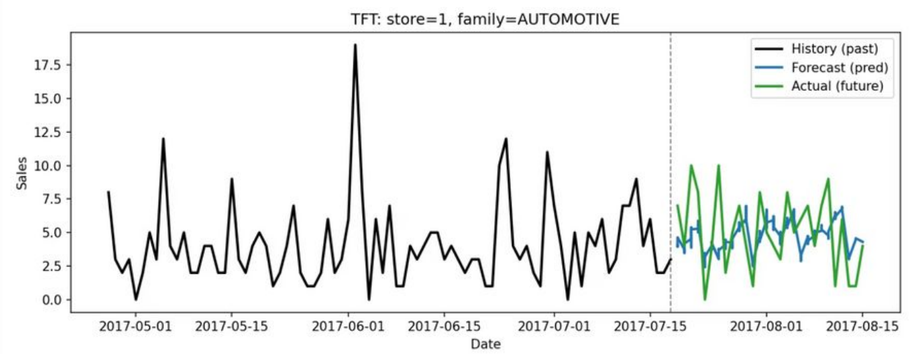
Temporal Fusion Transformer (TFT)



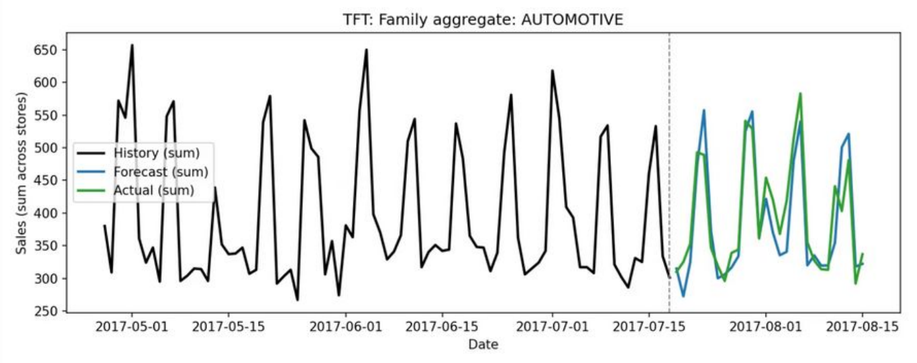
The Temporal Fusion Transformer (TFT) is a sophisticated deep learning model engineered for advanced multi-horizon time series forecasting. Diverging from simpler statistical approaches, TFT employs a cutting-edge neural network architecture to discern and capture intricate temporal patterns and dependencies within data.
This model offers significant advantages, including:

- **Adaptive Data Handling:** Robustly processes diverse time series data, incorporating both static and dynamic covariates.
- **Interpretable Predictions:** Utilizes attention mechanisms to provide insights into feature importance, enhancing model transparency.
- **Multi-Horizon Excellence:** Capable of accurately forecasting multiple future time steps concurrently.
- **Advanced Architecture:** Incorporates gated residual networks, variable selection networks, and a sequence-to-sequence layer for superior predictive performance.

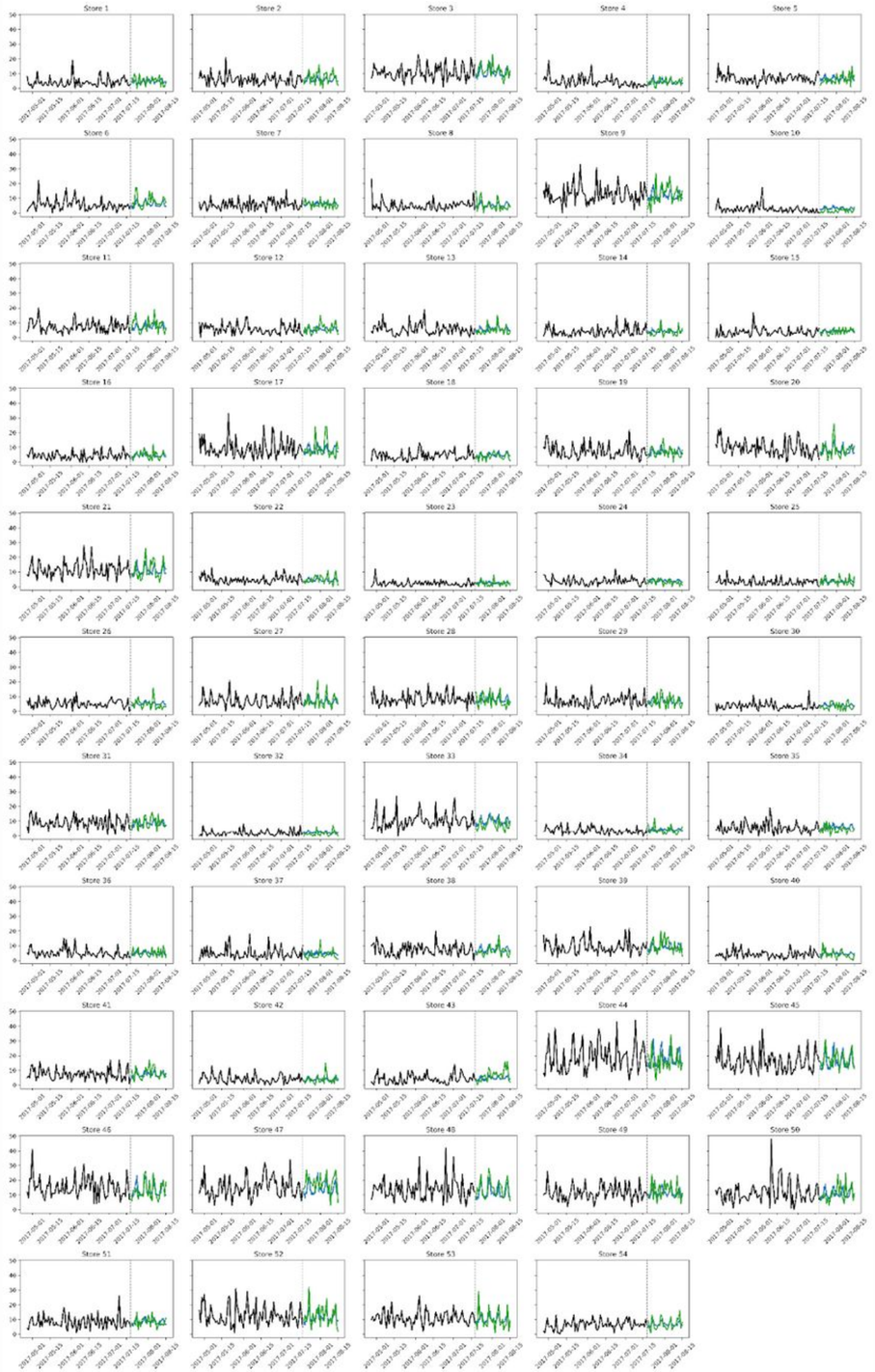
Temporal Fusion Transformer Results



Forecast specifically for the "Automotive" product type within Store 1



Aggregated forecasts for the "Automotive" product across all stores.



TFT Input Data

Feature Category	Feature Name
Past observed (9 total)	sales transactions daily oil price promotion day of the week month week of the year holiday workday
Known future (6 total)	promotion day of the week month week of the year holiday workday
Static (4 total)	store number product family store state store cluster

The TFT architecture requires three categories of input features: observed features from the past, known future inputs, and static features, as summarized in the table.

A sliding window mechanism provides the model with input-target pairs for each anchor time point t during training, validation and testing.

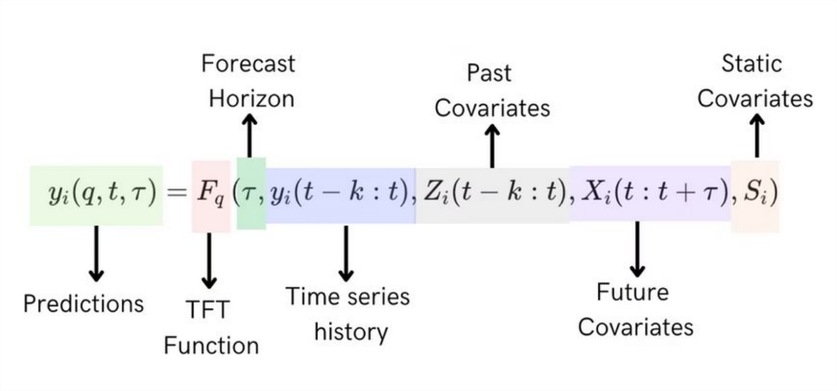


Image Reference: Medium

Model-Agnostic XAI:

Permutation Importance

Permutation Importance reveals critical features for forecast accuracy by measuring the increase in WAPE when a feature's values are shuffled.

- **Decoder (Future) Variables:** "onpromotion" is the most critical feature, causing the greatest increase in WAPE when shuffled.
- **Encoder (Historical) Features:** "sales" is the most important historical feature, while "transactions" and "onpromotion" have smaller but notable effects.
- **Static Features:** Store information and product metadata have a negligible impact on forecast performance.

Space	Variable	ΔWAPE
Decoder (future)	**onpromotion**	**0.1053**
	dow	0.0216
	month	0.0130
	weekofyear	0.0107
	is_holiday	0.0024
	is_workday	0
Encoder (historical)	sales	0.0280
	transactions	0.0020
	onpromotion	0.0018
	is_holiday	0.0002
	dow	0.0001
	is_workday	0
	weekofyear	-0.0001
	month	-0.0004
	dcoilwtico	-0.0016
Static	state	~0
	cluster	~0
	family	~0
	store_nbr	~0

Model-Intrinsic XAI: **Attention and Variable Selection Weights**

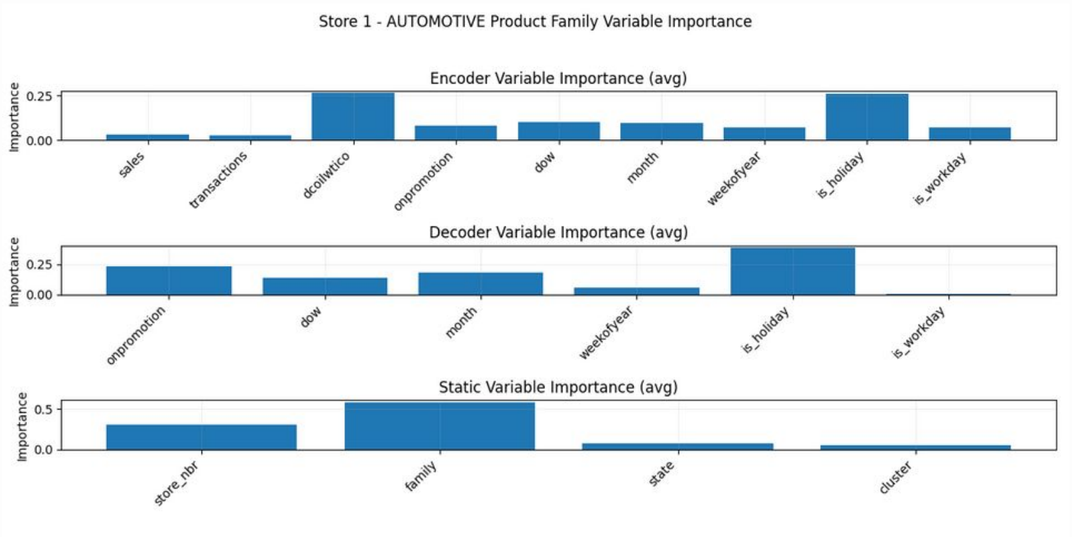
The TFT architecture provides built-in interpretability, offering direct insights into model decisions through its attention maps and variable selection networks.

Attention maps highlight which specific historical time points the model focuses on when making a prediction. They reveal the temporal segments the model prioritizes, helping us understand its focus over time.

The Variable Selection Network (VSN) assigns context-dependent importance weights to each input feature. It processes encoder (past), decoder (future), and static features, using gated residual networks. By applying a softmax function, these weights are normalized, resulting in a fused representation that clearly indicates the per-feature importance for any given prediction.

Key Model-Derived Variable Importances

- **High Impact:** Features like holiday, oil price, and family type
- **Moderate Impact:** 'onpromotion', day of the week, month, and store number
- **Low Impact:** Factors such as store state, cluster, sales, transactions, workday, and week of the year



Model Performance

Conducted experiments to evaluate four sets of key TFT architectural parameters:

- Hidden layer dimension
- Embedding dimension
- Number of attention heads
- LSTM hidden state size
- Number of stacked LSTM layers

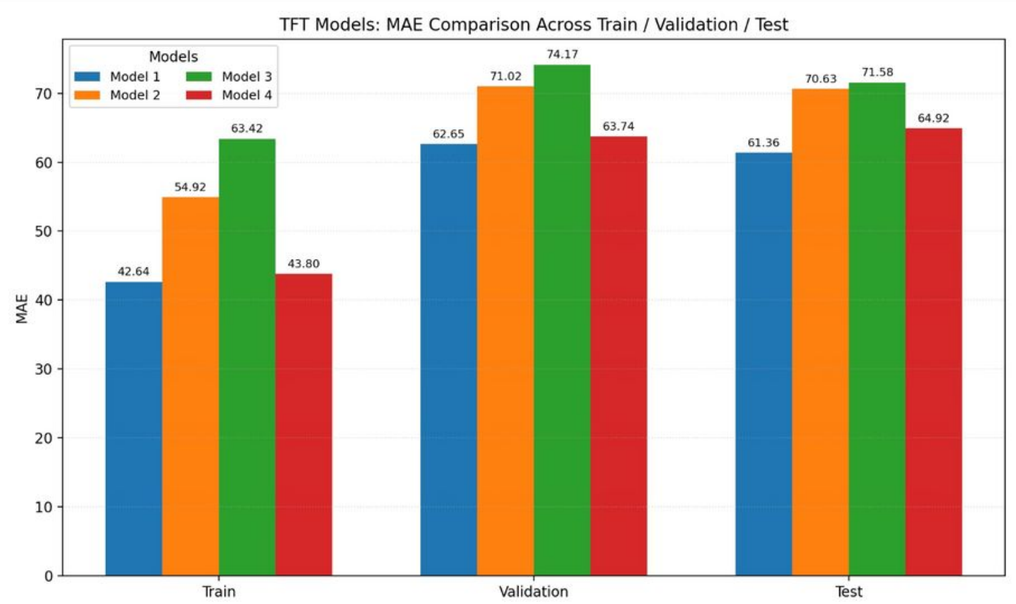
Best configuration identified:

- Hidden layer dimension: 128
- Embedding dimension: 64
- Attention heads: 4
- LSTM hidden state size: 64
- Stacked LSTM layers: 1

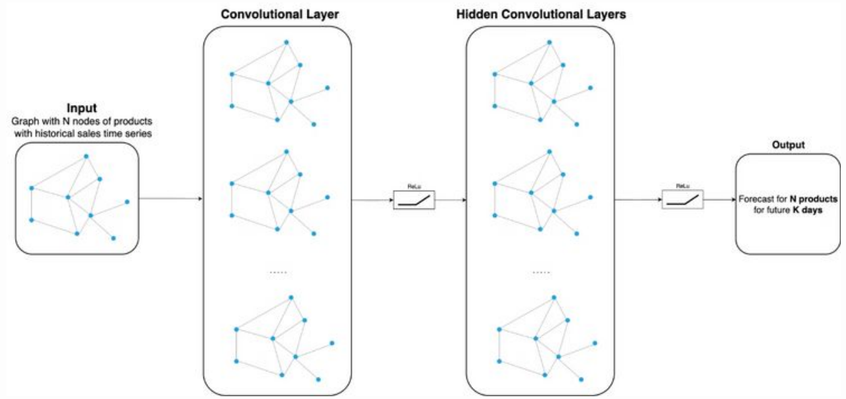
This optimal configuration achieved a Mean Absolute Error (MAE) of 61.36.

Key Architectural Parameters Evaluated

--hidden-dim	128	64	32	256
--d-model	64	32	16	128
--heads	4	2	2	8
--lstm-hidden	64	32	16	128
--lstm-layers	1	1	1	1
--dropout	0.1	0.1	0.1	0.3



Spatio-Temporal Graph Neural Network (STGNN)



The core of the STGNN architecture consists of spatio-temporal blocks, each composed of several key components:

Temporal Blocks

Capture sequential dependencies using dilated, causal, one-dimensional convolutions with causal padding. This ensures predictions maintain temporal causality by considering only current and historical data. Weights are shared across nodes.

Graph Convolutional Blocks

Process the underlying graph structure to capture spatial relationships. A first-order graph convolution is performed at each time-step: $H = \sigma(\hat{A} X W + b)$, where $\hat{A}$ is the normalized adjacency matrix, X are input features, W and b are learnable parameters, and σ is the ReLU activation.

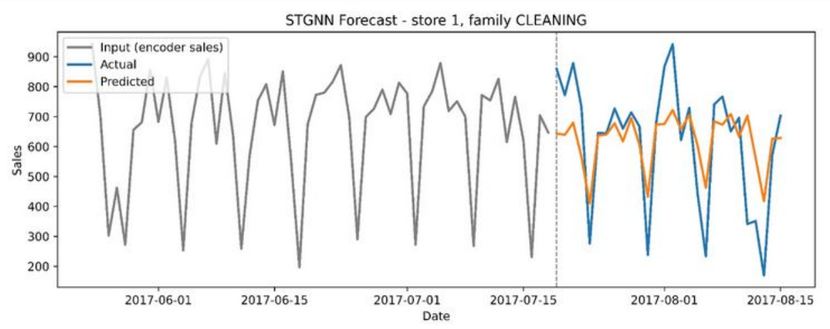
Linear Layers & ReLU

Linear layers map hidden representations to final output predictions, while the ReLU activation function introduces essential non-linearity into the model, allowing for complex pattern recognition.

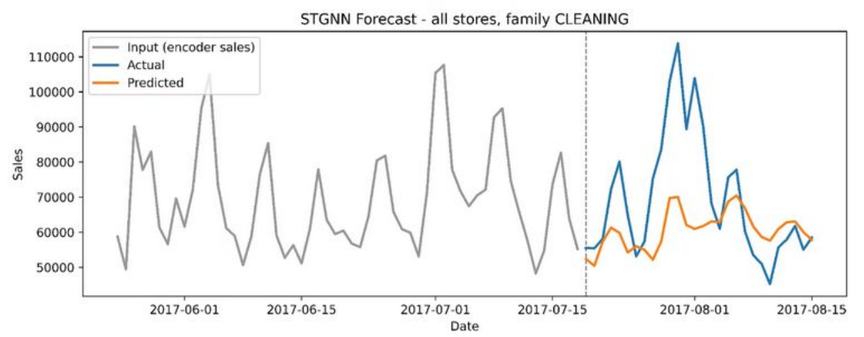
Spatio-Temporal Graph Neural Network

The STGNN architecture is designed for multi-horizon demand forecasting, adept at capturing both spatial and temporal dependencies. It provides an integrated approach to forecasting by modeling interconnected entities, such as product and store features, across multiple time steps.

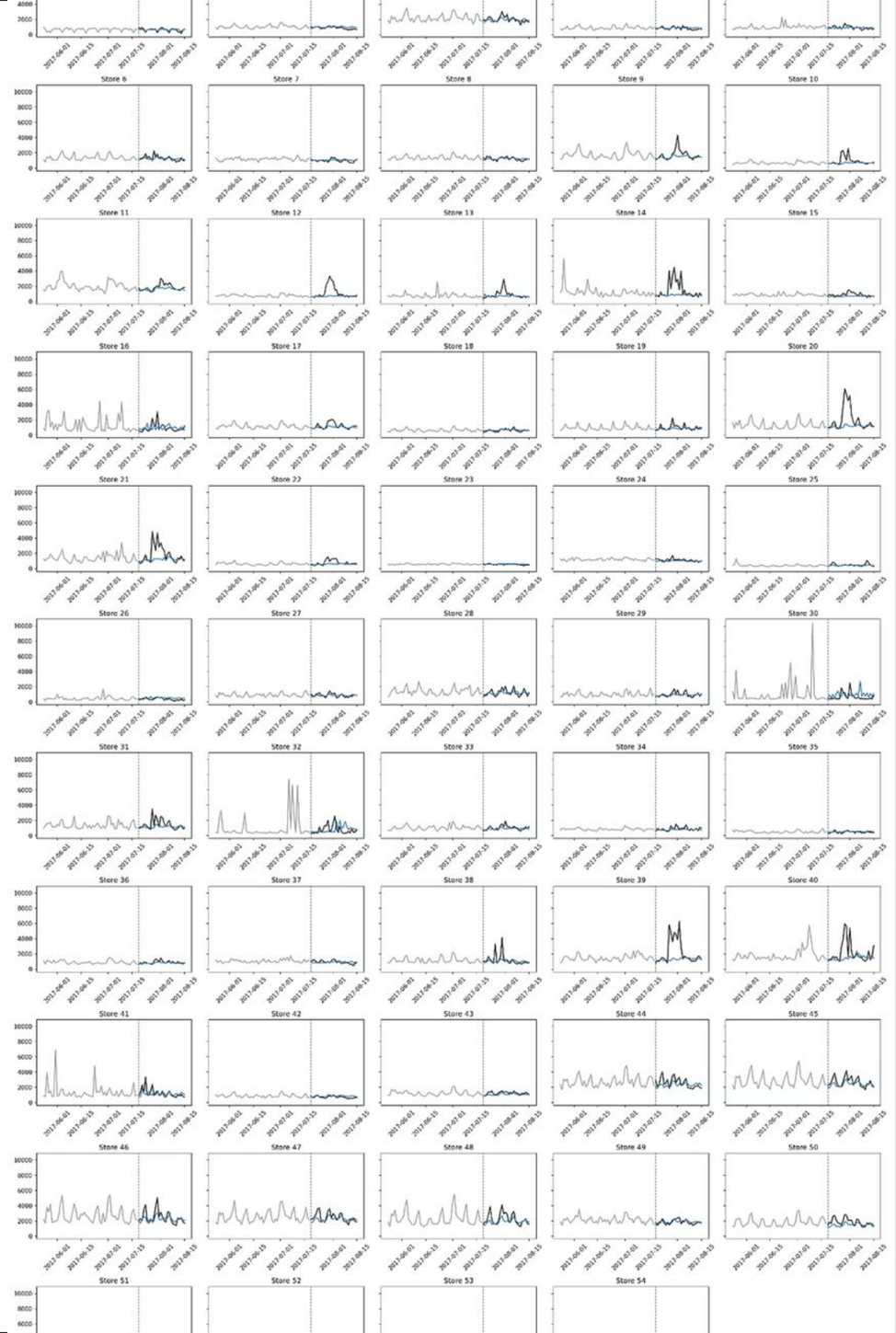
This model processes feature inputs alongside node information to deliver precise demand predictions for the next 28 days.



Forecast for the Cleaning product category in Store 1.



Forecast for Cleaning products across all Favorita stores in Ecuador.



STGNN Data

The spatio-temporal sliding window mechanism is employed to construct training, validation, and test sets for the STGNN model. In this framework, each node in the graph corresponds to a unique (store, family) pair, defined by assigning a unique node identifier to every combination of product family and store within the chronological data.

Dates (T)

The number of time steps or dates in the dataset.

Nodes (N)

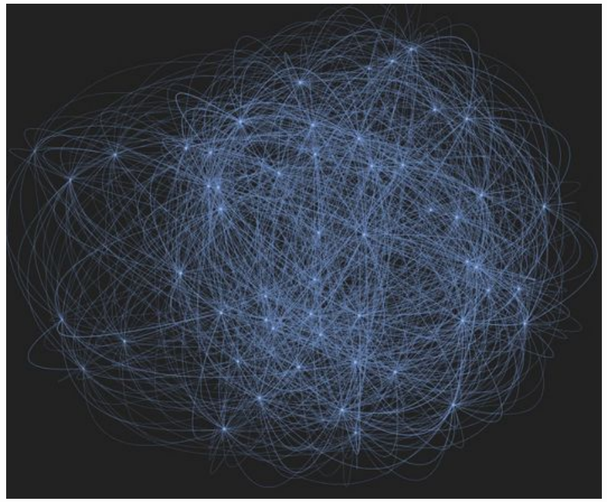
Each node represents a distinct (store, family) pair.

Features (F)

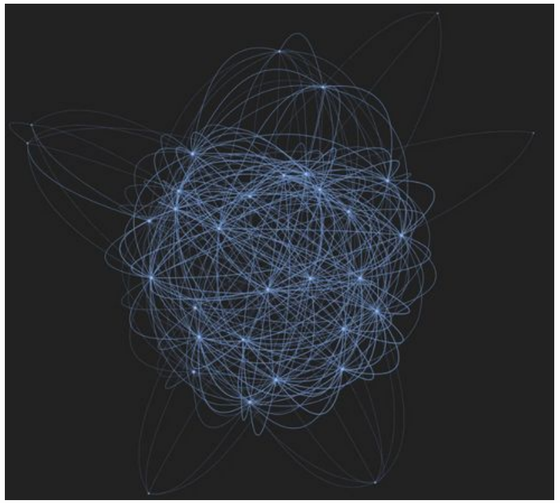
The number of distinct feature columns used for prediction.

Feature values for each date and node are organized into matrices, which are subsequently stacked into a three-dimensional tensor with shape [T, N, F].

The comprehensive feature set for each (store, family) pair includes: store_nbr, family, date, dow, month, weekofyear, id, sales, onpromotion, state, store_type, cluster, transactions, dcoilwtico, is_holiday, and is_workday.



Store Graph illustration



Product-Family Graph

Multi-relational graph

Kronecker-sum operation combines the store and family graph structures

$$A \oplus B = A \oplus I_b + I_a \oplus B$$

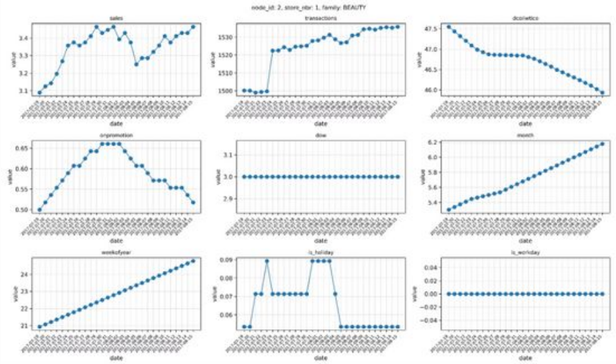
Store Graph

- **+0.5**: if both stores belong to the same cluster
- **+0.3**: if both stores are in the same state
- **+corr(transactions_a, transactions_b)**: Pearson correlation of the transaction time series

Family Graph

1. Pearson correlation coefficient is computed over a 60-day sales window for products (i) and (j)
1. If the correlation coefficient exceeds 0.3, an undirected edge is established between product (i) and (j), with the correlation value assigned as the weight

Feature-Level XAI: **SHAP** DeepExplainer



Feature	Mean absolute Shapley value
transactions	**0.6492**
dow	0.0033
sales	0.0031
weekofyear	0.0021
dcoilwtico	0.0020
onpromotion	0.0002
is_holiday	0.0002
month	0.0002
is_workday	0.0

Most influential feature—historical transactions are the primary driver of demand predictions.

- Transactions

Small but noticeable impact on predictions.

- Day of Week (dow)
- Sales
- Week of Year
- Oil Price Indicator (dcoilwtico)

Minimal contribution to model predictions.

- Onpromotion
- is_holiday
- month

Irrelevant Feature - No contribution to model’s decisions in the current setup.

Structure-Level XAI

Custom Edge-Mask Explainer

Decision Influencing Neighbors

Understanding the relationships of neighboring nodes is crucial for forecasting. Our custom edge mask optimization methodology provides clarity by:

Assigning Weights

Each neighboring connection receives a weight, representing its learned importance.

Optimizing with Gradient Descent

The edge mask is optimized using gradient descent, with L1 and entropy regularization, encouraging a sparse and interpretable mask.

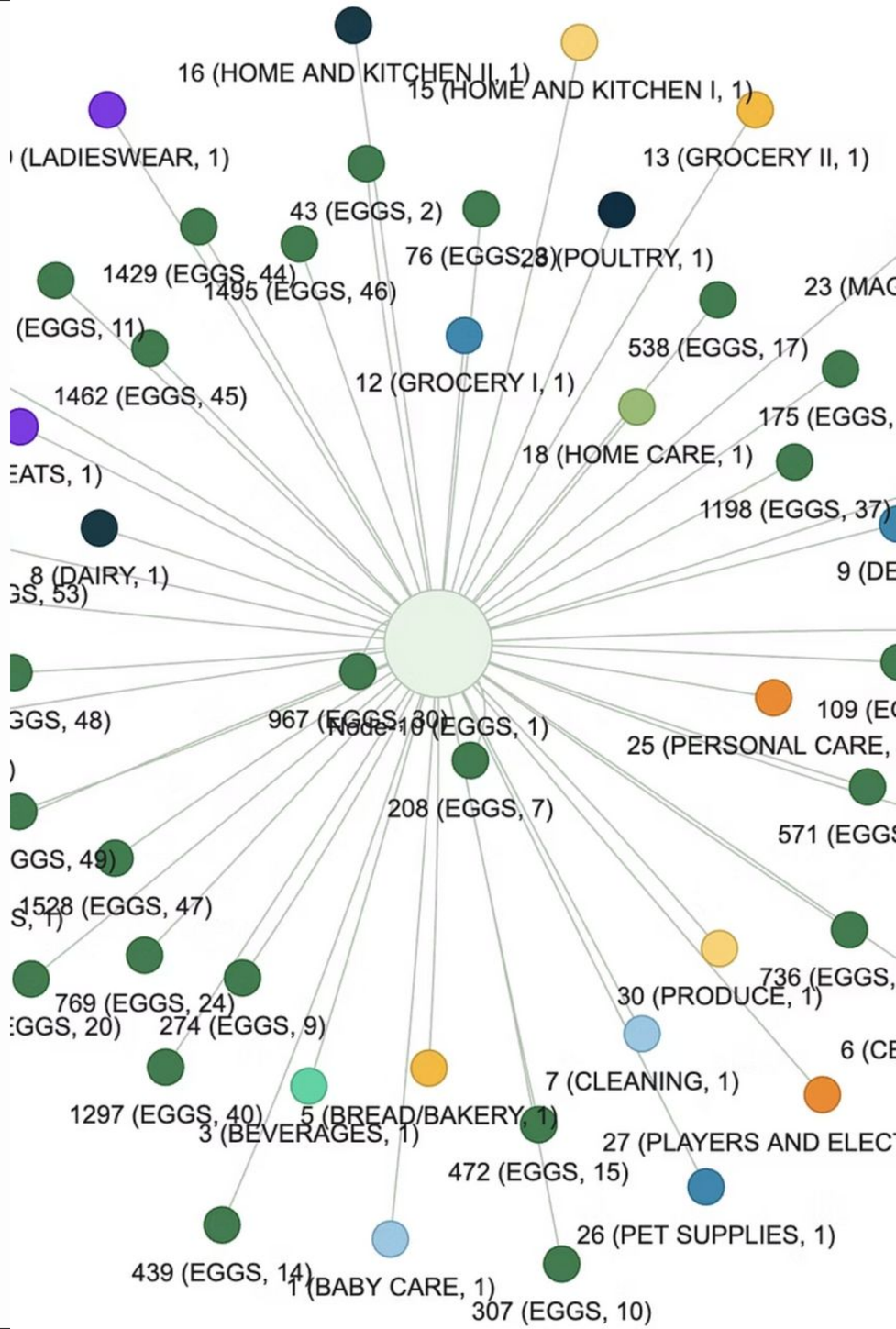
Quantifying Importance

The optimized mask quantifies the relative importance of each neighbor in forecasting the target node.

Identifying Key Influencers

Neighbors with larger mask values are interpreted as stores and products whose temporal patterns most influence the target node's prediction.

Figure 6.13 shows the node 10 (eggs product at store 1) neighbors that influence demand prediction for the product at store 10. See Appendix 10 for more details.



Performance

Training Setup

- Mini-batch size: 8
- Learning rate: 1×10^{-3}
- Dropout rate: 0.1 (to mitigate overfitting)
- Conditional quantile levels: 0.1, 0.5, 0.9 (probabilistic forecast range)
- Random seed set to 42 (ensures reproducibility)
- Early stopping: Patience of 5 epochs (prevents overfitting)

Computational Constraints

- Hidden size ≥ 128 leads to GPU A100 memory errors
- Hidden size set to the maximum feasible value given hardware limits

Test Performance

- Model 1: Achieves best test MAE = 61.31 (see Figure 6.14)

Model Configurations

Model 1

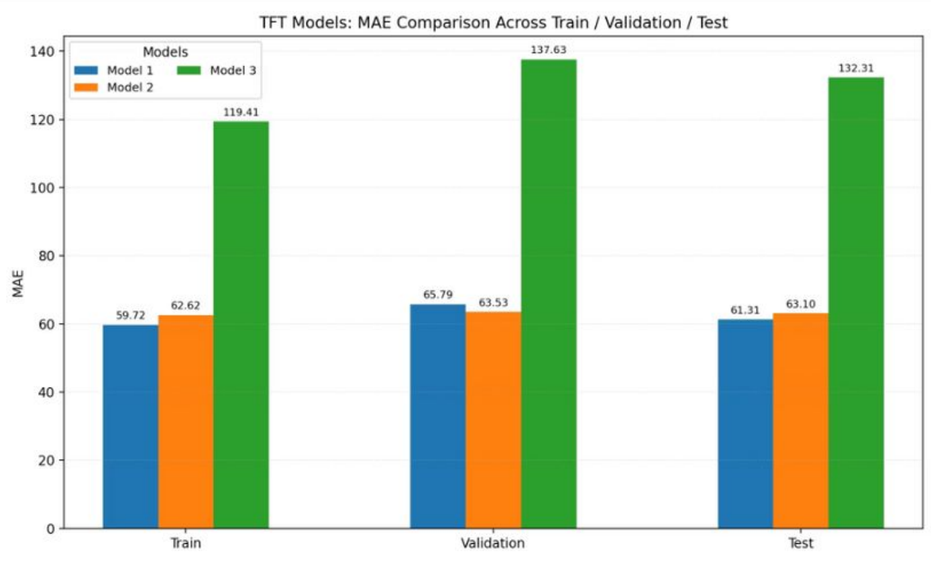
--hidden: 64
--blocks: 3
--kernel: 3

Model 2

--hidden: 32
--blocks: 3
--kernel: 3

Model 3

--hidden: 16
--blocks: 1
--kernel: 1



Hybrid Model Data

Key Features

Exploits both **relational** and **temporal patterns** in the input data for enhanced prediction accuracy.

Core Components

Composed of two primary components: (1) Relational Encoder (RE) and (2) Temporal Fusion Transformer (TFT).

Relational Encoding

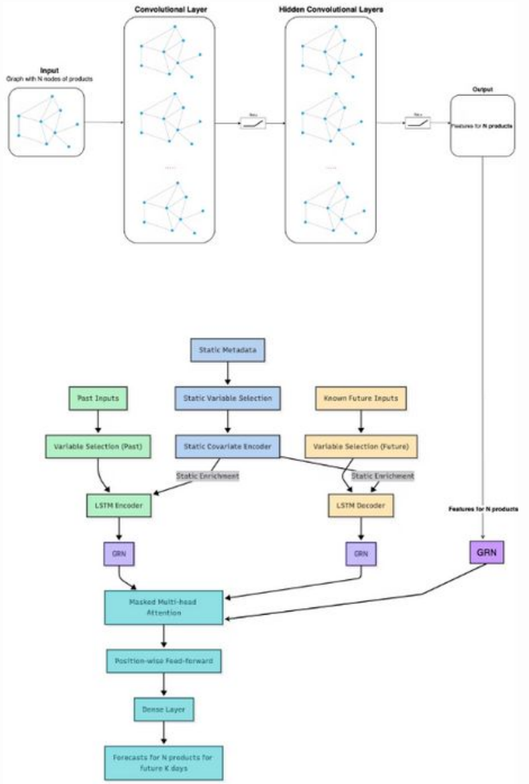
The RE employs a Graph Neural Network (GNN) to create node embeddings, effectively capturing the relational structure among product family and store entities.

Data Integration

The generated node embeddings are concatenated with static input features, and then passed to the TFT block as static covariates.

Embedding Generation

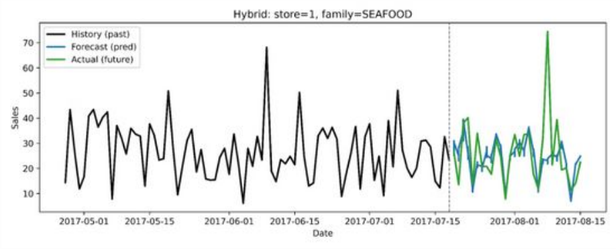
Node embeddings creation involves the computation of the Pearson correlation coefficient to identify significant relationships.



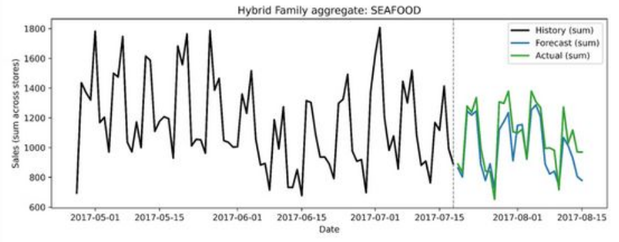
$$r = \frac{\sum(x - m_x)(y - m_y)}{\sqrt{\sum(x - m_x)^2 \sum(y - m_y)^2}}$$

Hybrid Model

The visualizations below demonstrate the effectiveness of our hybrid model in predicting seafood demand across different scopes.

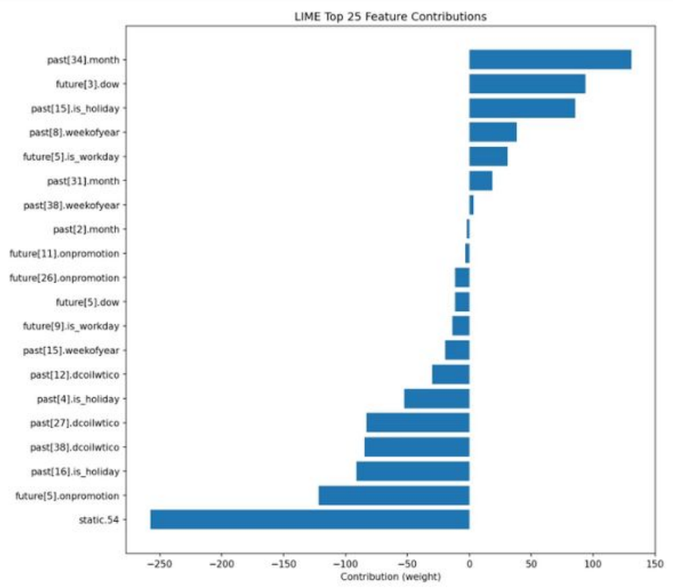


Store 1 Seafood Prediction



All Favorita Stores Prediction

Hybrid XAI: LIME



Feature	Weight
past[34].month	**130.8142**
future[3].dow	93.7328
past[15].is_holiday	85.4226
past[8].weekofyear	38.0508
future[5].is_workday	31.0180
past[31].month	18.6158
past[38].weekofyear	3.4062
past[2].month	-2.2726
future[11].onpromotion	-3.1617
future[26].onpromotion	-11.6736
future[5].dow	-11.7301
future[9].is_workday	-13.8870
past[15].weekofyear	-19.5332
past[12].dcoilwtico	-30.0725
past[4].is_holiday	-52.7784
past[27].dcoilwtico	-83.1999
past[38].dcoilwtico	-84.6973
past[16].is_holiday	-91.2767
future[5].onpromotion	-121.6794
static.54	**-257.7340**

Positive Influences

Temporal features significantly drive the model's prediction. Notably, past[34].month (130.81), future[3].dow (93.73), and past[15].is_holiday (85.42) are the most impactful contributors, highlighting strong dependencies on seasonality, day-of-week, and historical holiday effects. Other temporal and calendar-related features like past[8].weekofyear (38.06) and future[5].is_workday (31.02) also show positive influence.

Negative Influences

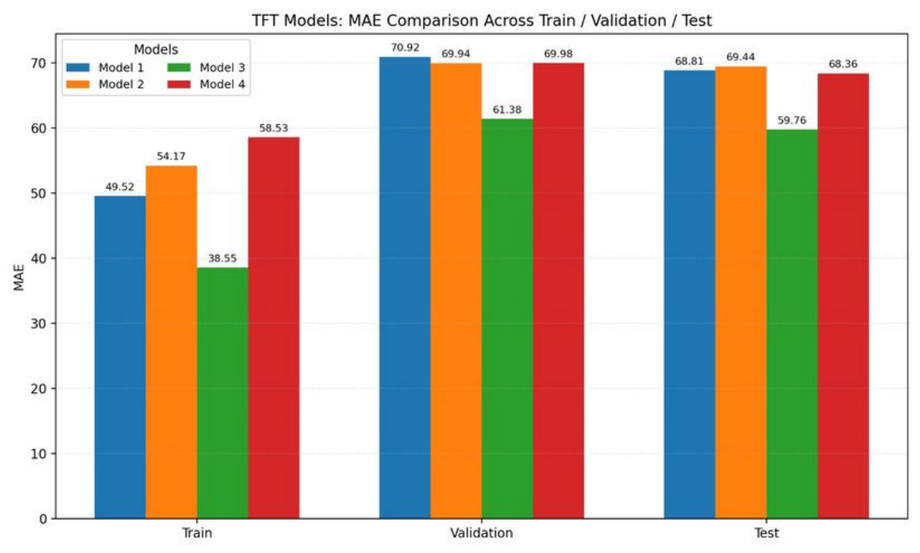
Conversely, several features markedly suppress the predicted value. Static.54 (weight = -257.73) has the most pronounced negative impact, indicating a strong association with reduced forecast outputs. Other features with negative weights include future[5].onpromotion (-121.68), past[16].is_holiday (-91.28), and dcoilwtico (oil prices) at various past intervals. These results suggest certain promotional periods, holidays, and economic indicators reduce demand prediction for the analyzed instance.

Overall LIME Analysis

This LIME analysis for store number 1 and product family AUTOMOTIVE reveals that the hybrid model's prediction is primarily driven by its temporal features, reflecting seasonality and calendar events. In contrast, specific static and external variables exert a substantial negative influence, providing critical insights into the model's decision-making for this particular prediction.

Hybrid Model Performance

Among the four evaluated configurations of hybrid model parameters, **Model 3** demonstrated the best performance, achieving the lowest MAE of 59.76 on the test set.



Model Configurations & Parameters

Parameter	Model 1	Model 2	Model 3 (Best)	Model 4
Hidden Dim	128	64	256	32
D-Mode/ Model	64	32	128	16
Heads	4	4	8	2
LSTM Hidden	64	32	64	32
LSTM Layers	1	1	3	1
GNN Hidden	64	32	128	16
GNN Embed	32	16	64	8

Model Performance Comparison

Model	Best Configuration	MAE	WAPE	SMAPE	Comments
ARIMA	order=(2,1,3)	4175.6045	0.1630	0.3366	Statistical baseline. Significantly higher error.
Linear Regression	Sklearn model fit	315.7444	0.8446	1.4174	Baseline Model. High error.
GNN	hidden=64 blocks=3 kernel=3	61.31	0.1286	0.5831	Comparable to best TFT.
Hybrid	hidden=256 emb=128 heads=8 LSTM=64×3 GNN=128	59.76	0.1253	0.5971	Best overall (lower MAE, WAPE).

Key Insights

TFT: Captures temporal sales patterns well, competitive MAE and WAPE.

GNN: Effective for structural and relational features; close results to TFT.

Hybrid: Outperforms both, leveraging both temporal & relational aspects.

Summary of Model Performance & Insights

Hybrid Model: Top Performer

The Hybrid model achieved the highest performance with a **59.76 MAE**, significantly outperforming statistical baselines (ARIMA, Linear Regression), and TFT, GNN.

Key Influencing Features

Features with the highest impact include **future promotion, holiday, oil price, product family type, transactions, month, day-of-week, and holidays**.

Less Impactful Features

Features with the least impact were **future workday, static metadata** (state, cluster, family, store number), and concatenated static features.

Unexpected Static Feature Impact

Despite the high performance of the hybrid model, XAI analysis revealed that **static features had the greatest negative impact** on the results.

Tuning Improves Accuracy

Experimental evaluation confirms that careful tuning of structural parameters (hidden layer dimensions, embedding sizes, attention heads, network depth) can **markedly improve predictive accuracy**.

Feasibility and Business Recommendations

Demand Planning

Store owners and managers can use the forecasts to proactively adjust inventory, optimize promotional campaigns, and prepare staffing according to expected demand fluctuations in the next 28 days.

Resource Allocation

Accurate multi-horizon forecasts allow for better allocation of capital towards inventory purchases, minimizing both stockouts and excess.

Strategic Marketing

Promotions and discounts can be strategically scheduled based on periods of forecasted lower demand, while high-demand periods can be leveraged for pricing optimization.

Explainability

The use of model interpretation techniques provides actionable insights into which factors drive demand, supporting evidence-based decision making for both short-term and strategic interventions.

Future Work

Scalability

The modeling framework can be extended to finer levels of granularity (individual SKUs) if higher-resolution data is available, further enhancing operational precision.

Research Objectives and Solutions

Research Objectives	How?
To evaluate and compare the forecasting accuracy of GNN, TFT, and hybrid GNN-TFT models.	• Developed each model using PyTorch. • Preprocessed and split the dataset. • Trained on Google Colab (A100, L4, T4 GPUs). • Performance measured by MAE, WAPE, and SMAPE.
To interpret model predictions by applying both model-specific and model-agnostic explainable AI (XAI) methods.	• **TFT**: Permutation Importance and intrinsic (built-in) XAI methods. • **GNN**: SHAP for feature-level, edge mask optimizer for neighbor-level explanations. • **Hybrid**: LIME for interpretability.
To find and implement architectural improvements, aiming to maximize predictive performance and operational value of models.	Trained models using various combinations of architectural parameters: • **TFT**: Hidden dim, model dim, heads, LSTM hidden, LSTM layers, dropout (4 combinations). • **GNN**: Hidden units, blocks, kernel sizes (3 combinations). • **Hybrid**: Hidden dim, model dim, heads, LSTM hidden, LSTM layers, GNN hidden, GNN embedding (4 combinations).

Thank you for your attention!

Q & A