Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting

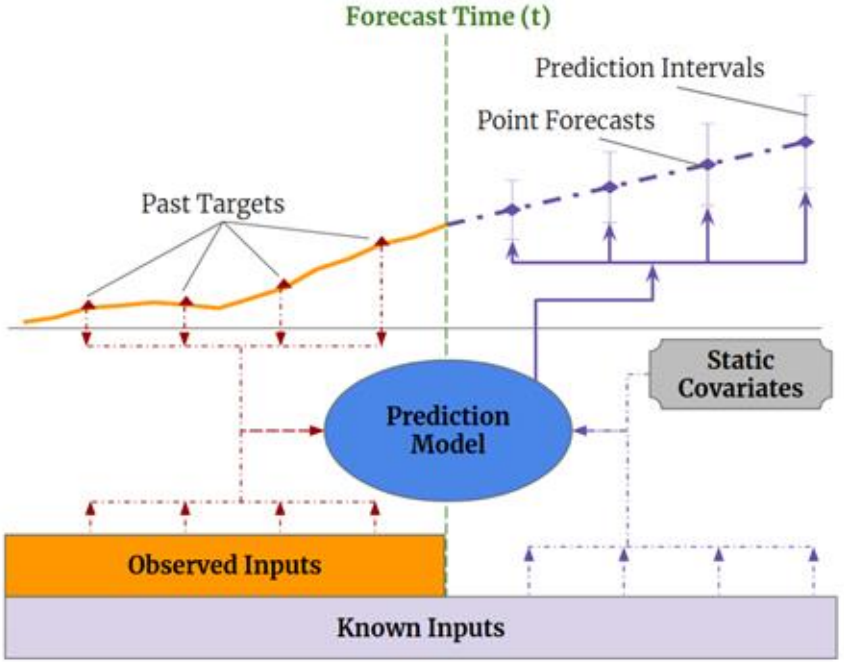
International Journal of Forecasting
Citation: 3333
Year: 2021

Introduction

Multi-Horizon Time Series Forecasting

Multi-horizon forecasting predicts multiple future time steps simultaneously.

- Unlike one-step prediction, it provides the **entire future trajectory**
- Enables better decision-making across time
  - Retail inventory planning
  - Healthcare treatment optimization
  - Economic forecasting



Real-World Challenge

Practical forecasting problems involve **heterogeneous inputs**:

- **Static covariates** (e.g., store location, patient ID)
- **Past-observed time-varying inputs** (e.g., historical demand, past glucose)
- **Known future inputs** (e.g., holidays, calendar effects)

These inputs:

- Interact in complex ways
- Are often poorly modeled by traditional methods

Limitations of Existing Deep Models

- RNN-based models assume all exogenous variables are known in the future
- Static covariates are often ignored or naively concatenated
- Transformer models focus on temporal attention but do not distinguish feature importance
- Most models are **black-box**, limiting interpretability

Problem definition

Setting

- Let there be **I entities** (e.g., patients, stores)
- For each entity i:
  - **Static covariates**

$$s_i \in \mathbb{R}^{m_s}$$

  - **Time-dependent inputs**

$$\chi_{i,t} = [z_{i,t}, x_{i,t}]$$

  - **Target variable**

$$y_{i,t} \in \mathbb{R}$$

At forecast time t, we predict:

$$y_{i,t+1}, y_{i,t+2}, \cdots, y_{i,t+\tau_{\max}}$$

We predict the entire future trajectory simultaneously.

Static features per entity.

Examples:

- Patient ID
- Sex
- Age
- Store location

They do NOT change over time.

Observed Inputs

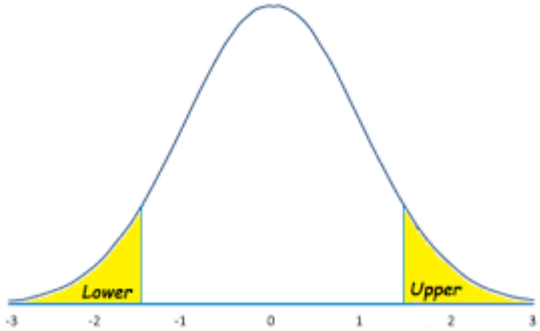
- Only known when measured
- Not available in the future
- Examples:
  - Past glucose
  - Past insulin
  - Past activity

Known Inputs

- Deterministic / pre-known
- Available in future
- Examples:
  - Hour-of-day
  - Day-of-week
  - Calendar features

Multi-Horizon Quantile Forecast

$$\hat{y}_i(q, t, \tau) = f_q(\tau, y_{i,t-k:t}, z_{i,t-k:t}, x_{i,t-k:t+\tau}, s_i)$$



Where:

- i: entity
- t: forecast time
- τ: prediction horizon
- q: quantile level
- k: lookback window

Inputs:

- Past target: $y_{i,t-k:t}$
- Past observed inputs: $z_{i,t-k:t}$
- Known inputs (past + future): $x_{i,t-k:t+\tau}$
- Static features: s_i

Output:

- Predicted q-th quantile at time $t + \tau$

Loss Function

🎯 Training Objective

TFT predicts multiple quantiles (e.g., 0.1, 0.5, 0.9).

It is trained using **quantile loss**:

$$QL(y, \hat{y}, q) = q(y - \hat{y})^+ + (1 - q)(\hat{y} - y)^+$$

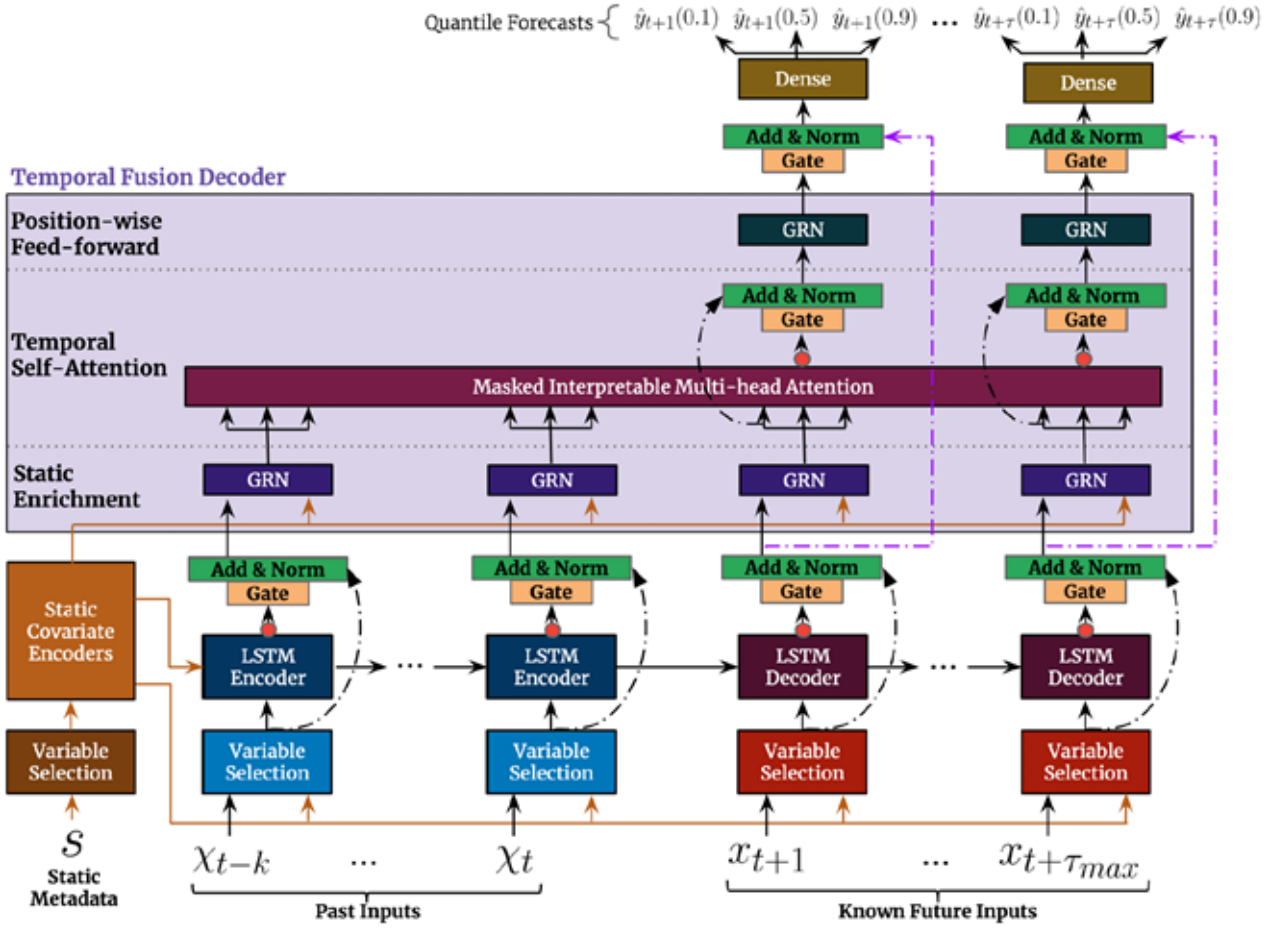
Where:

$$(x)^+ = \max(0, x)$$

Training loss = Sum over:

- All samples
- All forecast horizons
- All quantiles

$$q\text{-Risk} = \frac{\text{Total Quantile Loss}}{\text{Total Absolute True Values}}$$


Quantile Forecasts
$\hat{y}_{t+1}(0.1)$ $\hat{y}_{t+1}(0.5)$ $\hat{y}_{t+1}(0.9)$ ... $\hat{y}_{t+\tau}(0.1)$ $\hat{y}_{t+\tau}(0.5)$ $\hat{y}_{t+\tau}(0.9)$
Dense
Dense
Add & Norm
Gate
Add & Norm
Gate
Temporal Fusion Decoder
Position-wise Feed-forward
GRN
GRN
Add & Norm
Gate
Add & Norm
Gate
Temporal Self-Attention
Masked Interpretable Multi-head Attention
Static Enrichment
GRN
GRN
GRN
GRN
Static Covariate Encoders
Add & Norm
Gate
Add & Norm
Gate
Add & Norm
Gate
Add & Norm
Gate
LSTM Encoder
LSTM Encoder
LSTM Decoder
LSTM Decoder
Variable Selection
Variable Selection
Variable Selection
Variable Selection
Variable Selection
s
Static Metadata
χ_{t-k} ... χ_t
Past Inputs
x_{t+1} ... $x_{t+\tau_{max}}$
Known Future Inputs

Temporal Fusion Transformer Core Components

1 Gating Mechanisms

- Skip unnecessary parts of the architecture
- Provide adaptive depth and network complexity
- Allow flexibility across different datasets

2 Variable Selection Networks

- Select relevant input variables at each time step
- Dynamically control feature importance

3 Static Covariate Encoders

- Encode static features into context vectors
  Condition temporal dynamics on entity-specific information

4 Temporal Processing

- Capture both short-term and long-term dependencies
- **Sequence-to-sequence layer (LSTM)** → local processing
- **Interpretable multi-head attention** → long-range dependencies

5 Quantile Forecasting

- Produce prediction intervals
- Estimate the range of likely target values
  Enable uncertainty-aware multi-horizon forecasts

Gating Mechanism

$$z = W_2 a + W_3 c + b_2$$

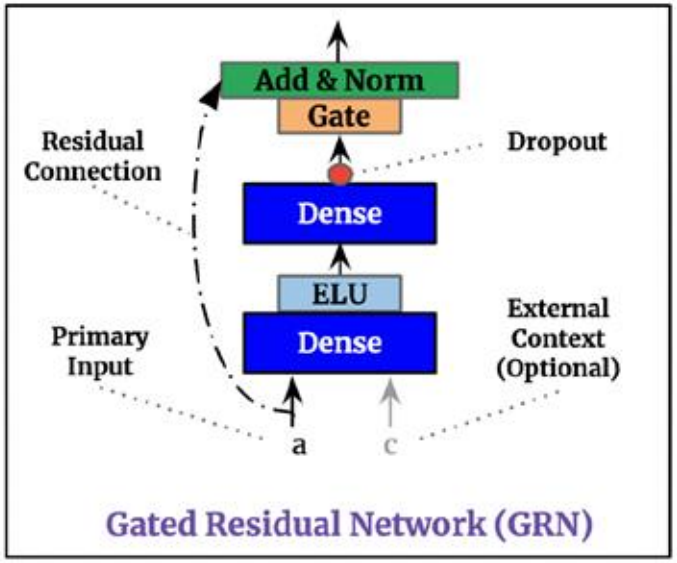
$$\eta_2 = ELU(z)$$

$$\eta_1 = W_1 \eta_2 + b_1$$

$$g = \sigma(W_4 \eta_1 + b_4)$$

$$s = W_5 \eta_1 + b_5$$

$$\text{Output} = \text{LayerNorm}\big(a + g \odot s\big)$$



Variable Selection Networks

⚙️ Feature Representation

Each variable is transformed into a vector of size d_{model}:

- Categorical variables → embeddings
- Continuous variables → linear transformation

For variable j at time t:

$$\xi_t^{(j)} \in \mathbb{R}^{d_{model}}$$

All variables are concatenated:

$$\Xi_t = [\xi_t^{(1)}, \ldots, \xi_t^{(m)}]$$

🔍 Variable Importance Weights

$$v_t = \text{Softmax}(\text{GRN}(\Xi_t, c_s))$$

- $v_t \in \mathbb{R}^m$
- Weights sum to 1
- c_s = static context vector

$$v_{\chi t} = [w_1, w_2, \ldots, w_m]$$

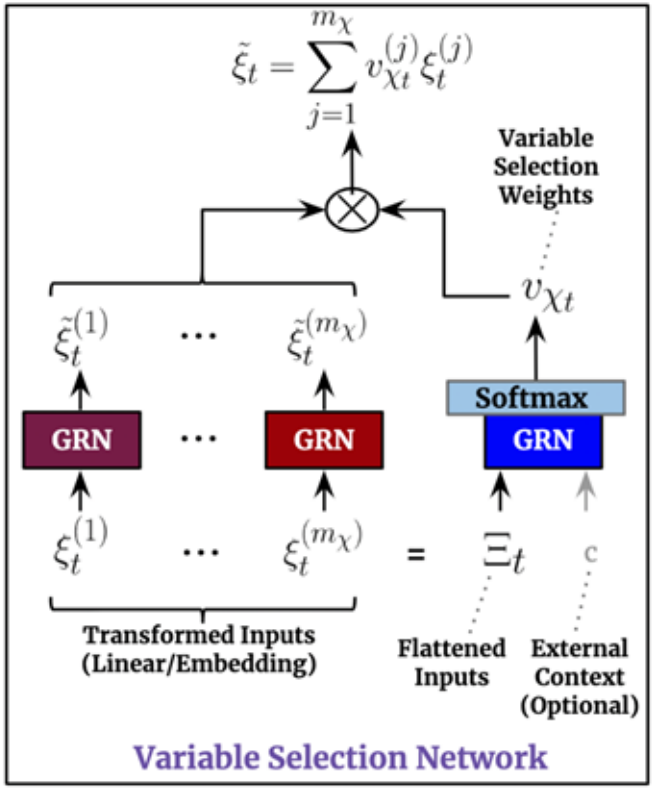
🔄 Feature-Specific Processing

Each variable passes through its own GRN:

$$\tilde{\xi}_t^{(j)} = \text{GRN}(\xi_t^{(j)})$$

➕ Final Representation

$$\tilde{\xi}_t = \sum_{j=1}^{m} v_t^{(j)} \tilde{\xi}_t^{(j)}$$



Static Covariate Encoders

different parts of the model need static information in different forms.

Context Vector	Used In	What It Influences
c_s	Variable Selection	Which features matter
c_c	LSTM Cell State Init	Long-term memory baseline
c_h	LSTM Hidden State Init	Short-term dynamics
c_e	Static Enrichment Layer	Interpretation of temporal patterns

🔷 Standard Self-Attention

Hidden states:

$$X \in \mathbb{R}^{N \times d_{model}}$$

$$Q = XW_Q, \quad K = XW_K, \quad V = XW_V$$

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_{attn}}}\right)V$$

- Compute similarity between Q and K
- Use weights to scale V

🔷 Standard Multi-Head Attention

For each head h:

$$Q^{(h)} = XW_Q^{(h)}, \; K^{(h)} = XW_K^{(h)}, \; V^{(h)} = XW_V^{(h)}$$

$$H^{(h)} = \text{Softmax}\left(\frac{Q^{(h)}K^{(h)T}}{\sqrt{d_{attn}}}\right)V^{(h)}$$

Final output:

$$\text{Concat}(H^{(1)}, \ldots, H^{(H)})W_O$$

⚠️ Problem:

Different value projections per head →

Attention weights ≠ true feature importance

🔷 TFT: Interpretable Multi-Head

Shared value projection:

$$V' = XW_V$$

Head-specific Q and K only:

$$Q^{(h)} = XW_Q^{(h)}, \quad K^{(h)} = XW_K^{(h)}$$

Average attention maps:

$$\tilde{A} = \frac{1}{H}\sum_{h=1}^{H} \text{Softmax}\left(\frac{Q^{(h)}K^{(h)T}}{\sqrt{d_{attn}}}\right)$$

Final output:

$$\tilde{H} = \tilde{A}V'$$

Thanks

Questions?