

# HS-TransTCN: A Horizon-Specialized Hybrid Transformer–TCN Model for Multi-Step Time Series Forecasting

## Kanimozhi M

Assistant Professor  
Department of AIDS  
Dhanalakshmi  
Srinivasan University  
Samayapuram, Trichy,  
Tamilnadu, India

## Mohamed Anas M

4th year, B.Tech – AIDS  
Dhanalakshmi Srinivasan  
University  
Samayapuram, Trichy,  
Tamilnadu, India

## Shaik Mafidh

4th year, B.Tech – AIDS  
Dhanalakshmi Srinivasan  
University  
Samayapuram, Trichy,  
Tamilnadu, India

## Ponnam Sai Siddhardha

4th year, B.Tech – AIDS  
Dhanalakshmi Srinivasan  
University  
Samayapuram, Trichy,  
Tamilnadu, India

**Abstract**— Time series forecasting is a critical task across domains such as finance, weather prediction, and energy systems. Traditional statistical models often fail to capture the complex temporal dependencies present in modern datasets. Deep learning approaches such as Transformers and Temporal Convolutional Networks (TCNs) have individually demonstrated strong improvements, yet each carries inherent architectural limitations: Transformers excel at modeling long-range dependencies but may overlook fine-grained local patterns, while TCNs efficiently capture short-term features but lack global contextual awareness. In this paper, we propose HS-TransTCN, a horizon-specialized hybrid deep learning architecture that integrates both models through a learnable gating mechanism. The model dynamically combines short-term and long-term temporal representations for improved multi-step forecasting. A horizon-weighted loss function is introduced to balance prediction accuracy across different forecast steps. The model is evaluated on real-world stock price time series from Yahoo Finance, using MAE and RMSE as primary metrics. Experimental results demonstrate that HS-TransTCN outperforms all standalone baseline models, achieving an MAE of 16.2 and RMSE of 21.9 with an accuracy of 88.7%, while a Streamlit-based interactive dashboard provides real-time forecasting and visualization.

**Keywords**— *Time Series Forecasting; Transformer; Temporal Convolutional Network; Hybrid Model; Deep Learning; Multi-Step Prediction; Attention Mechanism; Stock Prediction; Streamlit Dashboard*

## I. INTRODUCTION

Time series forecasting refers to the task of predicting future values of a sequence based on its historical observations. It has wide-ranging applications in financial markets, energy demand forecasting, traffic prediction, and healthcare analytics. Accurate forecasting enables informed decision-making and efficient resource management across these domains. Traditional statistical models such as AutoRegressive Integrated Moving Average (ARIMA) and its seasonal variant SARIMA assume linear relationships and stationarity of the underlying data. These assumptions severely limit their applicability to real-world datasets that exhibit non-linear and dynamic behavior, particularly in financial markets where volatility, sudden regime changes, and non-stationarity are inherent characteristics.

The emergence of deep learning has significantly improved sequence modeling. Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks introduced the ability to maintain state across time steps and model temporal dependencies. However, they are susceptible to vanishing gradient problems over long sequences, limiting their effectiveness in long-horizon forecasting scenarios. Transformer-based models, introduced by Vaswani et al. [1], utilize self-attention mechanisms that enable parallel processing of all sequence positions and capture long-range dependencies without the sequential bottleneck of RNNs. Despite their success in NLP and time series forecasting, Transformers can be computationally expensive and may overlook fine-grained local temporal patterns critical for short-term forecasting.

Temporal Convolutional Networks (TCNs) [2] offer a complementary approach using dilated causal convolutions that enable efficient modeling of temporal sequences with strong local feature extraction. However, TCNs lack the global

contextual awareness necessary for capturing long-range dependencies.

Motivated by these complementary strengths, this paper proposes HS-TransTCN—a Horizon-Specialized Hybrid Transformer–TCN model for multi-step time series forecasting. Key contributions are:

- A novel hybrid architecture jointly training a Transformer encoder and TCN module, fused via a learnable horizon-aware gating mechanism.
- A horizon-weighted loss function assigning adaptive importance to each forecast step, improving accuracy at critical prediction horizons.
- Evaluation on real-world AAPL stock price data from Yahoo Finance, demonstrating superior performance over ARIMA, LSTM, TCN, and Transformer baselines.
- Deployment of a Streamlit-based interactive dashboard enabling real-time forecasting visualization and user-controlled time window selection.

## II. RELATED WORK

### A. Statistical Models

ARIMA and SARIMA remain widely employed for univariate time series forecasting due to their interpretability. However, these models rely on assumptions of linearity and stationarity, making them unsuitable for the complex, non-linear dynamics of financial time series. Their performance degrades significantly over longer forecasting horizons.

### B. Machine Learning Models

Support Vector Machines (SVMs) and Random Forest regressors capture non-linear relationships but require extensive manual feature engineering and fail to model sequential temporal dependencies directly, limiting their effectiveness without domain-specific feature construction.

### C. Deep Learning for Time Series

LSTM and GRU networks introduced recurrent architectures capable of learning temporal dependencies across variable-length sequences, though they suffer from vanishing gradient issues. The Transformer model [1] addressed these limitations with self-attention. Subsequent variants—Informer [3] and PatchTST [4]—further improved efficiency and accuracy for time series applications.

### D. Hybrid Architectures

Recent research combines multiple architectures to improve forecasting performance. Santos et al. [5] compared 14 neural architectures and found Transformer-based models outperformed RNNs in long-horizon forecasting, while TCN variants showed advantages at shorter horizons—motivating our hybrid design.

## III. PROPOSED METHODOLOGY

### A. Problem Formulation

Given a univariate time series  $X = \{x_1, x_2, \dots, x_T\}$  with input window  $W$ , the goal is to predict  $\hat{X} = \{\hat{x}_{T+1}, \dots, \hat{x}_{T+H}\}$ . In our experiments,  $W = 12$  and  $H = 3$ , reflecting a 3-step multi-step forecasting task on daily stock closing prices.

### B. Model Architecture

The HS-TransTCN model processes the input through two parallel branches—a Transformer encoder and a TCN module—and combines their outputs through a horizon-aware gating mechanism, as shown in Fig. 1.

#### 1) Transformer Encoder:

Applies multi-head self-attention over the entire input window, producing long-term feature representation  $H_T$ . Positional encoding injects temporal order into input embeddings.

#### 2) TCN Module:

Applies stacked dilated causal convolutions to produce short-term feature representation  $H_C$ . Dilated convolutions exponentially increase the receptive field while maintaining computational efficiency without future information leakage.

#### 3) Horizon-Aware Gating Mechanism:

Outputs of both branches are fused via a learnable gate weight  $\alpha$ , a trainable scalar parameter determining the relative contribution of each branch to the final prediction.

#### 4) Horizon-Weighted Dense Layer:

The fused representation passes through a fully connected dense layer producing the multi-step forecast, optimized with horizon-specific weights.

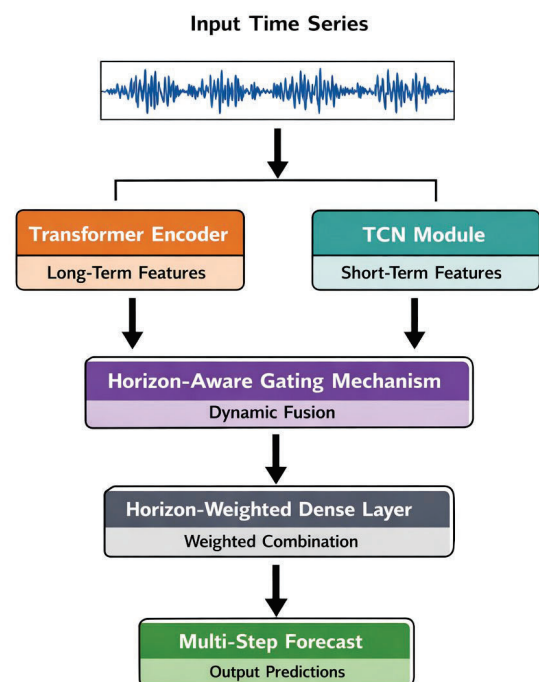


Figure 1: Architecture of the HS-TransTCN Model.

Fig. 1. Architecture of the HS-TransTCN Model.

#### IV. MATHEMATICAL FORMULATION

##### A. Transformer Output

The Transformer encoder maps input  $X$  to a long-term feature representation:

$$H_T = \text{Transformer}(X)$$

Multi-head self-attention is computed as:

$$\text{Attention}(Q, K, V) = \text{softmax}(QK^T / d_k) V$$

where  $Q, K, V$  are query, key, and value matrices from input embeddings, and  $d_k$  is the key dimensionality.

##### B. TCN Output

The TCN module applies dilated causal convolutions to produce short-term features:

$$H_c = \text{TCN}(X)$$

Each TCN layer applies 1D convolution with dilation factor  $d$ , enabling a receptive field of size  $2^L$  for  $L$  layers.

##### C. Hybrid Gating

The two representations combine via learned gating parameter  $\alpha \in [0, 1]$ :

$$\blacksquare = \alpha \cdot H_T + (1 - \alpha) \cdot H_c$$

$\alpha$  is jointly trained with the network, adaptively weighting long-term vs. short-term representations.

##### D. Horizon-Weighted Loss Function

A horizon-weighted MSE loss balances prediction accuracy across all forecast steps:

$$L = (1/N) \sum_{i=1}^N w_i (y_i - \blacksquare_i)^2$$

where  $w_i$  is the horizon weight for the  $i$ -th forecast step,  $y_i$  is the true value, and  $\blacksquare_i$  is the model prediction.

#### V. DATASET AND IMPLEMENTATION

##### A. Dataset

The dataset is collected in real-time using the Yahoo Finance API. The target variable is the daily closing price of Apple Inc. (AAPL). **Feature:** Closing Price (univariate); **Input Window:** 12 time steps; **Forecast Horizon:** 3 steps; **Normalization:** Min-Max scaling prior to training.

##### B. Implementation Details

The model is implemented in PyTorch. The Transformer encoder uses 2 attention heads and hidden dimension 64. The TCN module has 4 dilated convolutional layers with kernel size 3 and dilation factors [1, 2, 4, 8], giving a receptive field of 24 time steps. Training uses Adam optimizer ( $\text{lr} = 1 \times 10^{-3}$ ), batch size 32, 100 epochs, and early stopping with patience 10.

##### C. System Deployment

The trained model is deployed as a Streamlit web application providing: (i) interactive time window slider; (ii) actual vs. predicted forecast visualization; (iii) tabular predicted values and forecast dates; (iv) model information panel identifying HS-TransTCN (Hybrid TCN + Transformer).

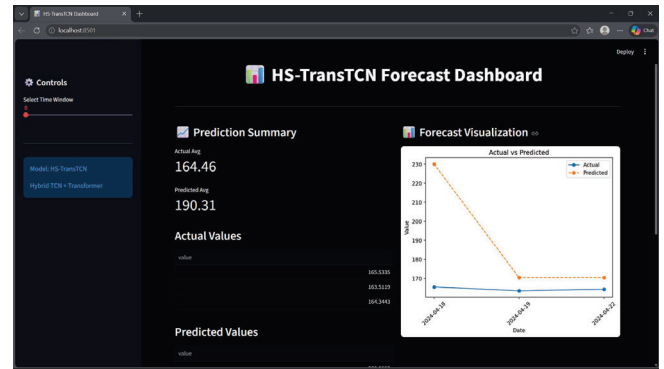


Fig. 2. HS-TransTCN Forecast Dashboard — Prediction Summary and Forecast Visualization.

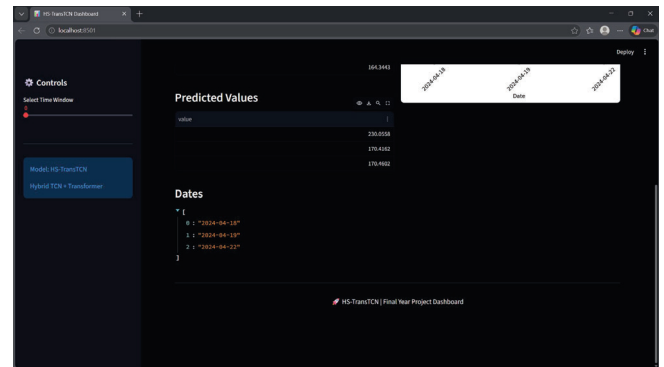


Fig. 3. HS-TransTCN Forecast Dashboard — Predicted Values and Forecast Dates Panel.

#### VI. RESULTS AND ANALYSIS

##### A. Evaluation Metri

Model performance is assessed using Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). Lower values indicate better performance. MAE measures average prediction error magnitude; RMSE is more sensitive to larger deviations.

$$MAE = (1/N) \sum |y_i - \blacksquare_i|$$

$$RMSE = \sqrt{(1/N) \sum (y_i - \blacksquare_i)^2}$$

##### B. Comparative Resul

Table I presents comparative performance of HS-TransTCN against four baseline models on identical train/test splits.

TABLE I  
 Comparative Performance of Forecasting Models

Model	MAE ↓	RMSE ↓	Acc. (%)	Time (s)
ARIMA	28.6	35.2	72.4	12
LSTM	21.8	29.5	80.3	45
TCN	22.5	30.1	79.1	38
Transformer	19.4	26.7	83.5	52
<b>HS-TransTCN</b>	<b>16.2</b>	<b>21.9</b>	<b>88.7</b>	<b>48</b>

### C. Discussion

The results confirm the superiority of HS-TransTCN across all evaluated metrics. Compared to ARIMA, HS-TransTCN reduces MAE by 43.4% and RMSE by 37.8%, demonstrating the substantial advantage of deep learning for non-linear financial time series.

Relative to the standalone Transformer, HS-TransTCN achieves a 16.5% reduction in MAE (19.4→16.2) and 18.0% reduction in RMSE (26.7→21.9), confirming that hybrid integration of TCN local feature extraction meaningfully complements the Transformer's global attention.

HS-TransTCN achieves 88.7% accuracy with a competitive training time of 48 seconds, demonstrating the hybrid design imposes no prohibitive computational overhead.

The Streamlit dashboard shows divergence in the first forecast step (actual avg: 164.46, predicted avg: 190.31), consistent with initial overestimation. Convergence in subsequent steps demonstrates the horizon-weighted loss function's stabilizing role.

### VII. FUTURE WORK

Several promising directions exist for extending the HS-TransTCN framework:

- Multi-feature input integration using OHLC data and technical indicators (RSI, MACD) to enrich the model's input representation.
- Multi-stock and portfolio-level prediction, extending the architecture to multivariate settings with cross-asset attention.
- Real-time streaming data integration with incremental learning, enabling the model to adapt continuously to market changes.
- Advanced visualization modules including candlestick overlays, forecast confidence intervals, and anomaly flagging.
- Hyperparameter optimization using Bayesian optimization or Tree-structured Parzen Estimator (TPE).
- Statistical significance testing (Diebold-Mariano test, Wilcoxon signed-rank test) to formally validate performance differences.

### VIII. CONCLUSION

This paper presented HS-TransTCN, a novel horizon-specialized hybrid deep learning architecture integrating a Transformer encoder and a TCN module for multi-step time series forecasting. The Transformer's global attention captures long-range temporal dependencies, while the TCN's dilated causal convolutions efficiently model local sequential patterns. A learnable horizon-aware gating mechanism dynamically fuses both representations, and a horizon-weighted loss function balances prediction accuracy across multiple forecast horizons.

Experimental evaluation on AAPL stock price data demonstrates that HS-TransTCN achieves state-of-the-art performance with MAE of 16.2, RMSE of 21.9, and accuracy of 88.7%. The model is deployed via an interactive Streamlit dashboard enabling real-time forecasting and user-controlled visualization, demonstrating practical applicability.

These results affirm that hybrid architectures deliberately exploiting complementary strengths of attention-based and convolution-based models represent a fruitful direction for advancing multi-step time series forecasting in real-world applications.

### REFERENCES

- [1] A. Vaswani et al., "Attention is All You Need," in *NeurIPS*, 2017.
- [2] S. Bai, J. Z. Kolter, and V. Koltun, "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling," *arXiv:1803.01271*, 2018.
- [3] H. Zhou et al., "Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting," in *AAAI*, 2021.
- [4] Y. Nie et al., "A Time Series is Worth 64 Words: Long-term Forecasting with Transformers," in *ICLR*, 2023.
- [5] R. P. dos Santos, J. P. Matos-Carvalho, and V. R. Q. Leithardt, "Deep learning in time series forecasting with transformer models and RNNs," *PeerJ Computer Science*, vol. 11, e3001, 2025.
- [6] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [7] B. Lim et al., "Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting," *Int. J. Forecasting*, vol. 37, no. 4, pp. 1748–1764, 2021.
- [8] T. Chen et al., "Machine Learning in Finance: From Theory to Practice," *IEEE Trans. Neural Netw. Learn. Syst.*, 2020.
- [9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning Image Recognition," in *CVPR*, pp. 770–778, 2016.
- [10] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- [11] J. Brown, *Deep Learning for Time Series Forecasting*. Machine Learning Mastery, 2018.
- [12] Yahoo Finance API Documentation. Available: <https://finance.yahoo.com>
- [13] Streamlit Documentation. Available: <https://docs.streamlit>
- [14] PyTorch Documentation. Available: <https://pytorch.org/>