NIFTY-50 AI Forecasting & Risk Analytics Platform

A Multimodal Deep Learning Approach with TCN and Risk-Aware Loss

Abstract

The Efficient Market Hypothesis (EMH) challenges the predictability of financial asset prices, ensuring that they reflect all available information. However, empirical stylised facts such as volatility clustering and leverage effects suggest that markets are not strictly efficient, opening avenues for predictive modeling. This paper introduces the 'NIFTY-50 AI Forecasting & Risk Analytics Platform', a novel multimodal deep learning framework designed for the Indian equities market. Departing from traditional single-source recurrent models, we propose a Temporal Convolutional Network (TCN) architecture that integrates OHLCV price data, technical indicators, macroeconomic variables, and news sentiment embeddings generated via DistilBERT. Our system replaces standard stock embeddings with a Gated Linear Unit (GLU) fusion mechanism and optimizes a hybrid objective function combining quantile regression for uncertainty quantification and binary cross-entropy for directional forecasting. Experimental validation on NIFTY-50 data (2015-2025) demonstrates the model's ability to capture long-range temporal dependencies and provide calibrated risk metrics, outperforming baseline LSTM models.

1 Introduction

Financial time-series forecasting remains one of the most challenging domains in predictive analytics due to the stochastic, non-stationary, and noisy nature of market data. The Efficient Market Hypothesis (EMH) posits that asset prices fully reflect all available information, rendering improved prediction impossible. However, the prevalence of market anomalies and behavioral biases contradicts the strong-form EMH, suggesting that deep learning models capable of capturing complex non-linear patterns can achieve superior predictive performance.

Existing literature has largely focused on single-modality models, typically utilizing historical price data fed into Long Short-Term Memory (LSTM) networks. While LSTMs are designed to handle sequential data, they often suffer from gradient vanishing problems over long sequences and lack the ability to process diverse data streams effectively. Furthermore, point estimates of future prices fail to capture the inherent uncertainty of financial markets, which is critical for risk management.

To address these limitations, we present a multimodal framework tailored for the NIFTY-50 index. Our approach draws inspiration from 'Stock2Vec', adopting a Temporal Convolutional Network (TCN) backbone to leverage its superior parallelization and receptive field properties. We distinctly diverge from prior work by:
1. Multimodal Fusion: Integrating technical indicators, macroeconomic regimes, and unstructured news sentiment (via DistilBERT) through a learnable gating mechanism.
2. Risk-Awareness: Replacing Mean Squared Error (MSE) with a quantile loss function to output confidence intervals (10th, 50th, 90th percentiles), enabling Value-at-Risk (VaR) estimation.
3. Market Specificity: Focusing on the idiosyncrasies of the Indian NIFTY-50 index rather than the S&P 500.

2 Related Work

Statistical models like ARIMA and GARCH have been the bedrock of econometrics, particularly for volatility modeling. However, their linear assumptions limit their efficacy in capturing the complex dynamics of modern financial markets. Machine learning approaches, including Support Vector Regression (SVR) and Random Forests, offered improvements but lacked the capacity to model temporal dependencies explicitly.

Recurrent Neural Networks (RNNs) and their variants, LSTMs and GRUs, became the standard for time-series tasks due to their memory cells. However, Bai et al. (2018) demonstrated that Temporal Convolutional Networks (TCNs) often outperform recurrent architectures in sequence modeling. TCNs employ causal dilated convolutions, allowing the

receptive field to grow exponentially with network depth, thus capturing long-range dependencies without the sequential processing bottleneck of RNNs.

Recent works have highlighted the importance of alternative data. Bollen et al. initially showed the correlation between Twitter sentiment and stock prices. Modern approaches utilize Transformer-based models like BERT to generate contextual embeddings from financial news. 'Stock2Vec' proposed a hybrid framework but relied on learning stock-specific embeddings. Our work extends this by directly fusing semantic text representations with quantitative market data.

3 Problem Formulation

We define the stock prediction task as a supervised learning problem combining regression and classification.
Let the market state at time t be represented by a multimodal tuple x_t = (p_t, m_t, s_t), where:

```
- p_t in R^d_p: Vector of OHLCV data and technical indicators.
- m_t in R^d_m: Vector of macroeconomic variables (e.g., volatility index).
- s_t in R^d_s: Sentiment embedding vector derived from news analytics.
```

Given a lookback window of size T, the input sequence is X_{t-T:t}. The goal is to learn a mapping function f(.) parameterized by theta that predicts the target variable y_{t+1}.

We predict two distinct targets:
1. Future Return Distribution (r^q): The predicted log-return at quantile q in {0.1, 0.5, 0.9}.
   r_{t+1} = ln(P_{t+1} / P_t)
2. Directional Probability (d): The probability that the price will close higher.

4 Data Representation

We utilize daily data for the NIFTY-50 index. To augment the raw Price-Volume signals, we compute a suite of technical indicators known to capture momentum and volatility:
- Momentum: Relative Strength Index (RSI, 14-day), MACD, MACD Signal.
- Trend: Simple Moving Averages (SMA-50, SMA-200), Exponential Moving Average (EMA-20).
- Volatility: Bollinger Bands (High/Low/Width) and Average True Range (ATR).
- Returns: Log-returns and rolling volatility (std dev of returns).

Unstructured textual data from financial news feeds is processed using DistilBERT, a distilled version of BERT that offers comparable performance with reduced computational cost. For each trading day, we aggregate news headlines and generate a semantic embedding s_t, capturing the prevailing market sentiment.

5 Model Architecture

The proposed Multimodal TCN architecture consists of three main components: parallel feature encoders, a multimodal fusion layer, and dual task-specific output heads.

A. Temporal Convolutional Encoders

We employ three parallel TCN encoders for price, macro, and text streams. Each TCN block utilizes:
1. Dilated Causal Convolutions: A 1-D convolution ensuring no information leakage from the future. Values depend only on inputs x_t, x_{t-1}, ..., x_{t-k*d}.

2. Residual Connections: facilitating gradient flow through deep networks.
3. Regularization: Dropout and Weight Normalization within each block.

B. Multimodal Fusion

The latent representations from the last time step of each encoder (h_p, h_m, h_s) are concatenated. We employ a Gated Fusion Mechanism to dynamically weight the modalities:

$$z = \text{Sigmoid}(W_f * h_{cat} + b_f) * h_{cat}$$

This allows the model to suppress noise (e.g., irrelevant news) and emphasize strong signals.

C. Output Heads

The fused vector z inputs into two fully connected networks:
- Quantile Head: Outputs values for q={0.1, 0.5, 0.9}.
- Probability Head: Outputs a scalar p in [0, 1] via Sigmoid representing the likelihood of a positive return.

6 Learning Objective

We minimize a composite loss function L that balances prediction accuracy with risk calibration:

$$L = \alpha * L_{quantile} + \beta * L_{BCE}$$

1. Quantile Loss: To model aleatoric uncertainty, we use the pinball loss. For quantile q and error e:

$$L_q = \max((q-1)e, qe)$$

2. Binary Cross-Entropy (BCE): Explicitly supervises the directional forecasting capability.

7 End-to-End Pipeline

The deployment pipeline ensures real-time operational capability:
1. Ingestion Layer: Fetches live market data via APIs and scrapes news headlines.
2. Feature Engineering: Computes technical indicators and generates DistilBERT embeddings.
3. Preprocessing: Scaling, sequence generation, and multimodal alignment.
4. Inference Engine: Executes the Multimodal TCN forward pass to generate return quantiles and directional probabilities.
5. Risk Analytics: Computes Value-at-Risk (VaR) from predicted 10th percentile returns and generates trading signals based on confidence thresholds.
This architecture ensures a holistic view of the market, balancing aggressive alpha generation with prudent risk management.