Multi-Stock Multimodal Multi-Resolution LSTM Fusion Network for Financial Prediction

Miquel Noguer i Alonso

Artificial Intelligence Finance Institute

September 15, 2025

Outline

1. Introduction

2. Model Architecture

3. Mathematical Specification

4. Training and Conclusion

The Challenge of Financial Forecasting

A Complex Data Environment

- Financial forecasting requires integrating diverse and asynchronous data sources.
- These sources include dense, high-frequency market data; sparse sentiment signals; and infrequent fundamental reports.
- Each data type has a unique temporal resolution, statistical properties, and rate of information decay.

Our Solution: The MS-MRFN

- We introduce the **Multi-Stock Multimodal Multi-Resolution Fusion Network** (MS-MRFN).
- It's a deep learning architecture designed specifically to handle these challenges.

Core Principles of the MS-MRFN

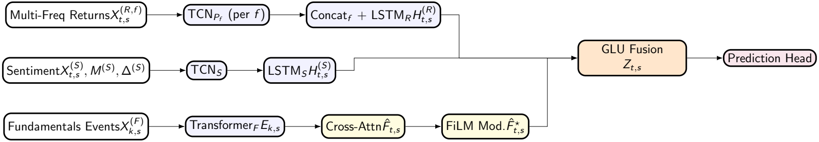
Three critical requirements form the foundation of our architecture:

1. **Respect native sampling frequencies:** Avoid information distortion from resampling by processing data at its natural timescale.
2. **Explicitly model data staleness:** Treat the absence of data and the age of information as explicit features, rather than using simple imputation. This allows the model to learn how to treat stale versus fresh information.
3. **Maintain strict time oredering:** Ensure predictions at time t only use information available at or before t to prevent information leakage from the future.

High-Level Architecture Overview

Structure

The MS-MRFN consists of three parallel, modality-specific encoders whose outputs are combined by a fusion module and a prediction head.



Multi-Freq Returns$X_{t,s}^{(R,f)}$

TCN$_{P_f}$ (per f)

Concat$_f$ + LSTM$_R H_{t,s}^{(R)}$

Sentiment$X_{t,s}^{(S)}, M^{(S)}, \Delta^{(S)}$

TCN$_S$

LSTM$_S H_{t,s}^{(S)}$

Fundamentals Events$X_{k,s}^{(F)}$

Transformer$_F E_{k,s}$

Cross-Attn$\hat{F}_{t,s}$

FiLM Mod.$\hat{F}_{t,s}^{\star}$

GLU Fusion
$Z_{t,s}$

Prediction Head

Encoder 1: Multi-Frequency Returns

Goal

Capture market dynamics across multiple time horizons simultaneously.

- **Temporal Convolutional Networks (TCNs):**
  - A separate TCN is used for each frequency (e.g., daily, weekly).
  - Each TCN uses a dilation rate (P_f) matched to its frequency's period.
  - This ensures the receptive field aligns with the data's period (e.g., a weekly TCN looks at data points from $t, t-5, t-10, \ldots$).
- **Fusion & Temporal Modeling:**
  - The outputs from all frequency-specific TCNs are concatenated.
  - An LSTM processes this combined vector to model interactions between the different frequency components.

Encoder 2: Sparse Sentiment Data

Goal

Process sparse and irregularly-timed sentiment signals while accounting for data staleness.

- **Augmented Input:**
  - Instead of just raw values, the input vector is augmented.
  - It includes the sentiment score ($X^{(S)}$), a binary presence mask ($M^{(S)}$), and a staleness feature ($\Delta^{(S)}$) representing time since the last update.
- **Processing:**
  - This augmented sequence is fed into a TCN-LSTM stack.
  - This allows the model to learn to rely heavily on new information but discount its value as it becomes older.

Encoder 3: Fundamental Events (Part 1)

Goal

Encode discrete, low-frequency fundamental reports (e.g., quarterly earnings) and align them with daily market data.

- **Event Encoding with a Transformer:**
  - The sequence of a stock's fundamental reports is treated like a sentence.
  - A Transformer Encoder processes this sequence to create contextualized embeddings for each report.
  - This captures the financial history and trends, rather than treating each report in isolation.

Encoder 3: Fundamental Events (Part 2)

- **Daily Alignment with Cross-Attention:**
  - To use this event-level data on a daily basis, a cross-attention mechanism is employed.
  - The daily market state $(H^{(R)}_{t,s})$ from the Returns Encoder acts as the "query".
  - It attends to the sequence of fundamental report embeddings (the "keys" and "values").
  - This dynamically selects the most relevant historical fundamental data for the current market context.

Encoder 3: Fundamental Events (Part 3)

- **Staleness Conditioning with FiLM:**
  - The attended fundamental vector ($\hat{F}_{t,s}$) represents *what* information is relevant.
  - We use a Feature-wise Linear Modulation (FiLM) layer to condition this on *how* relevant it is based on its age.
  - The FiLM layer uses the time since the last report and time until the next report to generate a gain (η) and bias (β).
  - This modulation can amplify or suppress features based on their position in the earnings cycle.

Fusion and Prediction

Goal

Adaptively combine the representations from the three encoders to make a final prediction.

- **Concatenation:**
  - The output representations from the Returns ($H^{(R)}$), Sentiment ($H^{(S)}$), and Fundamentals ($\hat{F}^{\star}$) encoders are concatenated.
- **Gated Linear Unit (GLU) Fusion:**
  - A GLU is used to fuse these representations.
  - It employs a gating mechanism to dynamically control the information flow, selecting the most useful features for the prediction.
- **Prediction Head:**
  - The final fused vector ($Z_{t,s}$) is passed to a linear layer to produce the forecast (e.g., classification or regression).

Inputs and Data Structure

For each stock s and day t:

- **Multi-Frequency Returns:** $X_{t,s}^{(R,f)} \in \mathbb{R}^{d_f}$ for each frequency $f \in \mathcal{F}$.
- **Sentiment Data:** A tuple containing:
  - Features: $X_{t,s}^{(S)} \in \mathbb{R}^{d_S}$
  - Presence Mask: $M_{t,s}^{(S)} \in \{0, 1\}$
  - Staleness: $\Delta_{t,s}^{(S)} \in \mathbb{R}_{\geq 0}$ (days since last update)
- **Fundamental Events:**
  - A sequence of reports $X_{k,s}^{(F)} \in \mathbb{R}^{d_F}$ at times $\tau_{k,s}$.
  - Daily staleness indicators: $\Delta_{\text{since}}^{(F)}$ and $\Delta_{\text{to}}^{(F)}$.

Returns Encoder Equations

1. **Frequency-Matched TCN Layer:**

$$Y_{t,s}^{(f)} = \mathrm{ReLU}\left(\sum_{i=0}^{k-1} W_i^{(f)} \cdot X_{t-i\cdot P_f,s}^{(R,f)} + b^{(f)}\right)$$

   The full stack produces $Z_{1:T,s}^{(R,f)} = \mathrm{TCN}_{P_f}(X_{1:T,s}^{(R,f)})$.

2. **Concatenation and LSTM:**

$$Z_{t,s}^{(R)} = \bigoplus_{f\in\mathcal{F}} Z_{t,s}^{(R,f)}$$

$$H_{1:T,s}^{(R)} = \mathrm{LSTM}_R(Z_{1:T,s}^{(R)})$$

   This yields the final market state representation $H_{t,s}^{(R)}$.

Sentiment Encoder Equations

1. **Augmented Input Vector:**

$$\tilde{X}_{t,s}^{(S)} = \left[X_{t,s}^{(S)};\ M_{t,s}^{(S)};\ \phi(\Delta_{t,s}^{(S)}) \right]$$

where $\phi(\cdot)$ is a positional encoding for the staleness feature.

2. **TCN-LSTM Stack:**

$$Z_{1:T,s}^{(S)} = \mathrm{TCN}_S(\tilde{X}_{1:T,s}^{(S)})$$
$$H_{1:T,s}^{(S)} = \mathrm{LSTM}_S(Z_{1:T,s}^{(S)})$$

This produces the daily sentiment state representation $H_{t,s}^{(S)}$.

Fundamentals Encoder Equations (1/2)

1. **Event Encoding (Transformer):**

$$E_{1:K_s,s} = \mathrm{Transformer}_F(X^{(F)}_{1:K_s,s}) \in \mathbb{R}^{K_s \times d_H}$$

   This creates contextual embeddings for each of the K_s reports.

2. **Cross-Attention (Query, Key, Value):**

$$\begin{aligned} Q_{t,s} &= W_Q H^{(R)}_{t,s} \quad \text{(from Returns Encoder)} \\ K_{k,s} &= W_K E_{k,s} \quad \text{(from Transformer)} \\ V_{k,s} &= W_V E_{k,s} \quad \text{(from Transformer)} \end{aligned}$$

Fundamentals Encoder Equations (2/2)

3. **Cross-Attention (Output):**

$$\hat{F}_{t,s} = \sum_{k \text{ s.t. } \tau_{k,s} \leq t} \alpha_{t,k,s} V_{k,s}$$

where $\alpha_{t,k,s}$ are masked softmax attention weights.

4. **FiLM Modulation:**

$$\begin{aligned}[\eta_{t,s}, \beta_{t,s}] &= \mathrm{MLP}(\Gamma_{t,s}) \\ \hat{F}^{\star}_{t,s} &= \eta_{t,s} \odot \hat{F}_{t,s} + \beta_{t,s}\end{aligned}$$

where $\Gamma_{t,s}$ is a vector of staleness indicators. This produces the final fundamentals representation $\hat{F}^{\star}_{t,s}$.

Fusion and Prediction Equations

1. **Concatenation:**

$$U_{t,s} = [H^{(R)}_{t,s}; H^{(S)}_{t,s}; \hat{F}^{\star}_{t,s}]$$

2. **GLU Fusion:**

$$Z_{t,s} = (W_z U_{t,s} + b_z) \odot \sigma(W_g U_{t,s} + b_g)$$

   where σ is the sigmoid function and $\odot$ is element-wise multiplication.

3. **Prediction Head (Binary Classification):**

$$\hat{p}_{t,s} = \sigma(w^\top Z_{t,s} + b)$$

Training Protocol

Preventing Lookahead Bias

- The model is trained using a rigorous walk-forward validation scheme.
- The dataset is split into chronological training, validation, and unseen test sets.
- This process can be repeated on a sliding window to assess performance across different market regimes.

Time ordering Enforcement

- All components are designed to be strictly time oredered.
- TCNs use padding, and the cross-attention mechanism is masked to prevent attending to future fundamental reports.

Conclusion and Future Work

Summary

- MS-MRFN offers a principled architecture for multimodal financial forecasting in a multi-stock environment.
- It systematically addresses the challenges of asynchronous and heterogeneous data by:
  - Respecting native data frequencies.
  - Explicitly modeling data staleness.
  - Strictly enforcing time ordering.
- Its modular design provides a powerful framework for synthesizing market, sentiment, and fundamental data.

Future Work

- Future work could explore incorporating graph-based structures to model inter-stock relationships (e.g., supply chains, sector correlations) directly within the architecture.