Cross-Modal Temporal Fusion for Financial Market Forecasting

Yunhua Pei[a], **John Cartlidge**[b,*], **Anandadeep Mandal**[c], **Daniel Gold**[d], **Enrique Marcilio**[d] and **Riccardo Mazzon**[d]

[a]School of Computer Science, University of Bristol, Bristol, UK
[b]School of Engineering Mathematics and Technology, University of Bristol, Bristol, UK
[c]Business School, University of Birmingham, Birmingham, UK
[d]Stratiphy Limited, London, UK

Abstract. Accurate forecasting in financial markets requires integrating diverse data sources, from historical prices to macroeconomic indicators and financial news. However, existing models often fail to align these modalities effectively, limiting their practical use. In this paper, we introduce a transformer-based deep learning framework, Cross-Modal Temporal Fusion (CMTF), that fuses structured and unstructured financial data for improved market prediction. The model incorporates a tensor interpretation module for feature selection and an auto-training pipeline for efficient hyperparameter tuning. Experimental results using FTSE 100 stock data demonstrate that CMTF achieves superior performance in price direction classification compared to classical and deep learning baselines. These findings suggest that our framework is an effective and scalable solution for real-world cross-modal financial forecasting tasks.

1 Introduction

Forecasting financial markets is a challenging and high-risk task, with implications for investment strategies, risk management, and economic policy. The primary objective is to accurately predict the prices of financial assets in order to generate potential profits. In this context, stock prediction, a crucial aspect of financial markets, has gained increasing attention over the past few years.

The Efficient Market Hypothesis (EMH) [12, 13] suggests that market efficiencies place limitations on the ability to consistently generate excess returns. In weak form efficiency, it is assumed that asset prices incorporate all information in past prices, making technical analysis ineffective; in semi-strong form, prices are assumed to reflect all public information, including historical prices, news, earnings reports, and economic data, therefore also rendering fundamental analysis ineffective; and in strong form efficiency prices are assumed to reflect both public and private information, making even insider trading ineffective. However, the EMH is controversial and there is ample empirical evidence of market inefficiencies [2], suggesting that it is possible to predict prices for excess returns.

The financial industry has been exploring prediction models since the early twentieth century [7], continuously advancing these technologies through substantial financial investments. Traditional quantitative approaches mainly rely on historical time series data to forecast stock movements [25, 33]. However, with the development of deep learning, more recent efforts have explored approaches to decompose complex market dynamics [44, 23] and capture stock interdependencies through attention mechanisms [36, 8].

Lately, advances in Natural Language Processing (NLP) have enabled the integration of unstructured textual data to enhance prediction models. For example, news [16, 24, 41] and social media content [17, 34] can be analyzed for sentiment to generate a scoring matrix of positive-negative signals for each stock, which is then incorporated as a new input feature. These event-driven methods focus on extracting valuable patterns from event information for stock prediction.

Although prior work has achieved some success in stock prediction, three open challenges remain: (1) *Heterogeneous data integration* – existing methods [e.g., 25, 44] tend to crudely aggregate multi-frequency inputs (e.g., quarterly reports, daily price series, and real-time news) without aligning their temporal dependencies, which can lead to loss of signal or spurious correlations; (2) *Superficial interpretability* – existing use of attention mechanisms [43, 20] fails to disentangle specific drivers (e.g., GDP trends vs. news) or provide actionable insights, making it difficult for practitioners to understand or trust predictions; (3) *Inflexible training paradigms* – characterized by rigid architectures with inefficient retraining and hyperparameter optimization [47, 6], existing methods have limited ability to rapidly adapt to volatile market conditions.

To address these challenges and solve the core problem of aligning and extracting value from diverse financial signals, we propose CMTF, a cross-modal temporal fusion unified framework that (i) integrates multimodal data, (ii) ensures forecasting interpretability, and (iii) automates training schemes for rapid iteration with hyperparameter tuning. The CMTF framework provides valuable insight for practitioners dealing with diverse data types, offering guidance on how to effectively handle and select variable features during the feature engineering process. Furthermore, to improve training efficiency, it introduces hyperparameter search rules to the attention-based model, enabling faster convergence, faster iterations, and optimized performance. The main contributions of this work can be summarized as follows:

- We propose a multimodal tensor representation that integrates structured data (historical market data and macro-index) and unstructured data (news sentiment and financial reports); enabling systematic alignment of heterogeneous temporal and event-driven

[*] Corresponding Author. Email: john.cartlidge@bristol.ac.uk.

signals for stock market forecasting.

- We design a sparse tensor interpretation framework that leverages Lasso regression for feature selection and attention mechanisms to prioritize cross-modal interactions (e.g., linking event-driven trends to modality-specific price movements); ensuring interpretable and actionable predictions.
- We conduct extensive experiments on real-world stocks from the FTSE 100 index. The experimental results demonstrate that CMTF outperforms a suite of baselines in forecasting the next trading day close price, with average improvements of 1.52% in precision, 30.38% in recall, and 0.17 in F1 score for the classification task.

The contributions and findings of this work have informed the development of Stratiphy's emerging applications. Stratiphy is a wealth management platform for retail investors to build active portfolios using industry-leading trading strategies and risk management tools.[1] The CMTF framework is being prototyped as an emerging application to develop new investment strategies for business and retail customers. These advances are essential to the continued success of Stratiphy and offer the social benefits of better financial investment and risk management for all. Code and data availability[2].

2 Literature Review

2.1 Multimodality

Recent advances in multimodal learning have enabled the fusion of structured and unstructured data for financial forecasting. [19] introduced temporal fusion transformers to jointly model static covariates (e.g., sector metadata) and dynamic time series data, but their fixed temporal alignment struggles with low-frequency earnings reports. To address this, [21] introduced a cross-modal transformer to align daily X (formerly Twitter) data with historical price trends, but their fixed temporal windows ignore intermodal frequency mismatches. A notable advancement is [39], which introduced the momentum transformer, which combines technical indicators with attention mechanisms to capture momentum-driven market regimes. However, like [32], which fused numerical and textual datasets using cross-modal attention, these methods lack mechanisms to synchronize different granularity data with momentum shifts. Furthermore, despite their predictive performance, these models fail to provide clear explanations for their final results, limiting their interpretability and practical use in real financial decision-making.

2.2 Financial Time-Series Forecasting

Modern financial time series forecasting increasingly leverages hybrid architectures. [45] proposed Informer, a transformer variant optimized for time series prediction, which reduces the huge inference cost. For high-frequency trading data, such as the limit order book, [18] designed a reinforcement learning framework with volatility-sensitive rewards. The efficiency of different model topologies has also been explored: some works, such as [15, 42, 23], use a graph-based topology, where stocks are represented as nodes with static and dynamic connections; in contrast, traditional multi-time series models follow a sequential topology, treating each stock as an independent time series without explicitly constructing their relationships. Recent work by [40] introduced Autoformer, which leverages auto-correlation to decompose market trends and seasonal effects; however, the model remains restricted to single-modal input and will not adapt to sudden market changes (e.g., a black swan event). Finally, most of these models directly use pre-processed data and do not address the complexities of processing multimodal unstructured data.

2.3 Interpretability

For industrial applications, in particular, interpretability is a key requirement of financial forecasting models, as commercial vendors and regulators require actionable insights into model decisions. Post hoc explainability tools, such as attention maps [19] and saliency methods [30], have been introduced for deep learning models, but the explanations provided lack economic foundations. New deep learning architectures have also been introduced to improve interpretability. For example, to disentangle patterns, [20] proposed a Temporal Routing Adaptor with optimal transport, which learns distinct trading patterns and assigns stocks to patterns using dynamic routing. For multimodal settings, [43] designed a Domain-Adaptive Neural Attention Network that aligns news sentiment trends with sector-specific price movements via cross-modal attention; however, although domain adversarial training improves robustness to distribution shifts, interpretability is reduced by masking attribution to specific modalities.

3 Preliminary

3.1 Notation

To formalize our methodology, here we define the key notation used for our CMTF model. Let t denote the current time step, and let T be the set of time steps until t, so $t \in \{1, ..., T\}$. Let D represent the number of input features after multimodal fusion, and N correspond to the number of target stocks for prediction. Then, the tensor $\mathcal{X} \in \mathbb{R}^{T \times D}$ is the final input tensor containing all encoded features over time, and the model can make predictions for the next day's $(t+1)$ close prices $\hat{P}_{t+1}$ for N stocks, where $i \in \{1, ..., N\}$ is each stock. Finally, Z^h, Z^m, Z^n, and Z^r represent the structured tensors derived from *historical data*, *macro index*, *news*, and *financial reports*, respectively. We use this notation consistently throughout the remainder of this paper.

3.2 Task Definition

In this work, we tackle the classification task of financial market forecasting, in which the objective is to predict the direction of the movement of the stock price: whether the stock price will go up or down the next day. At each time step, t, we define binary classification labels for the true direction of change. Given that the closing prices in our dataset do not remain the same on two consecutive days, we adopt a straightforward binary classification approach:

$$\textit{true\_direction}_{t+1}^{i} = \begin{cases} 1, & \text{if } p_{t+1}^{i} - p_{t}^{i} > 0 \\ 0, & \text{if } p_{t+1}^{i} - p_{t}^{i} < 0 \end{cases} \tag{1}$$

and, similarly, the predicted direction of change:

$$\textit{pred\_direction}_{t+1}^{i} = \begin{cases} 1, & \text{if } \hat{p}_{t+1}^{i} - p_{t}^{i} > 0 \\ 0, & \text{if } \hat{p}_{t+1}^{i} - p_{t}^{i} < 0 \end{cases} \tag{2}$$

[1] Stratiphy can be found at https://www.stratiphy.io.

[2] Code is available at https://github.com/PEIYUNHUA/CMTF; for data access, please contact Stratify.

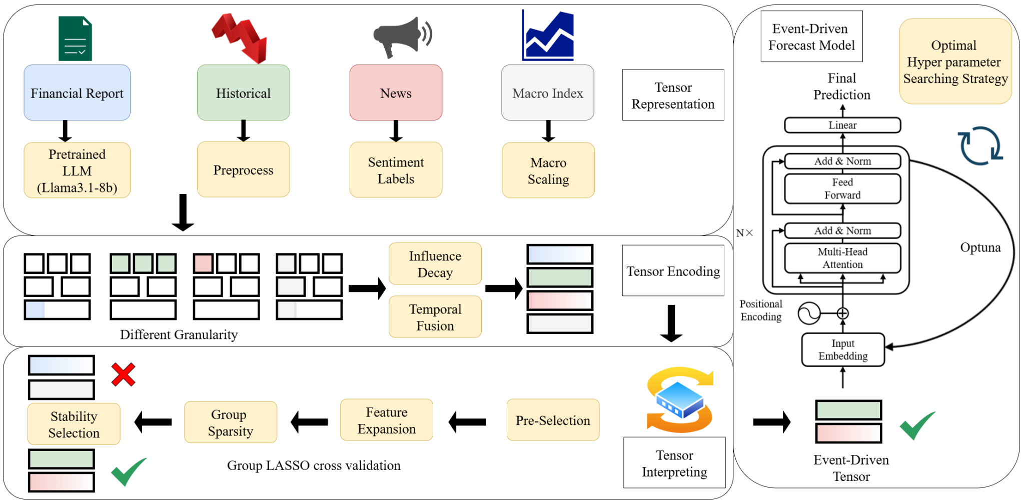

Figure 1: Overview of proposed CMTF. The framework integrates multimodal data (historical data, macro index, news, and financial reports). It employs Tensor Representation (extract tensor representation from unstructured data), Tensor Encoding (scale and preprocess the collected tensor), Tensor Interpretation (select important tensors), and a Transformer-based forecasting model (apply the optimal training scheme).

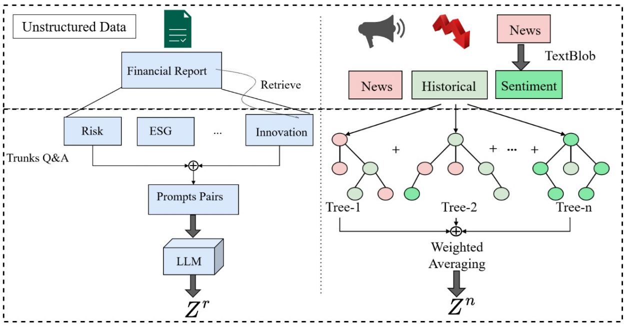

Figure 2: Tensor representation pipeline of financial reports and news.

where p_t^i ($p_t^i \in P_t$) represents the closing price of stock i at time t, and p_{t+1}^i refers to the closing price of stock i at time $t+1$. The classification model aims to predict whether the price of the stock will increase or decrease, with labels 1 (increase) and 0 (decrease).[3] The objective is to minimize the binary cross-entropy loss between the predicted and true directions.

4 Methodology

An overview of the CMTF framework is presented in Fig. 1. It consists of four components: tensor representation; tensor encoding, tensor interpretation; and a transformer-based forecasting model. Here, we introduce each of these components in detail.

4.1 Tensor Representation

This work uses two types of data: *structured data*, consisting of predefined numerical values that can be used directly for model training; and *unstructured data*, including textual information, which requires pre-processing before training. Here, we specifically focus on textual data for further processing. At the stage of tensor representation, we aim to transform different data types into tensor representations suitable for further training in Fig. 2.

Specifically, we consider two types of unstructured textual data: *news* and *financial report*. To represent these data, we introduce two complementary approaches: CatBoost for extracting classification tensors Z^n for news; and a Large Language Model (LLM) to generate rating value tensors Z^r for financial reports.

To extract the tensor representation from unstructured news data, we implement a version of CatBoost's gradient-boosting optimized decision trees [26]. The loss function shows as:

$$\arg\min_{\theta} \sum_{m=1}^{M} \Bigg[\underbrace{\sum \ell(z_i^n, F_{m-1}(\mathbf{x}_i) + f_m(\mathbf{x}_i))}_{\text{Boosted Loss}} \\ + \underbrace{\lambda ||f_m||^2}_{\text{L2 Reg}} + \underbrace{\gamma(\mathbf{x}_i)}_{\text{Encoding Stabilizer}} \Bigg] \tag{3}$$

where $\mathbf{x}_i$ combines processed text signals and market technicals, including news sentiment, intraday price dynamics, and historical volatility; and z_i^n represents the binary classification labels of next day price movements. Both $\mathbf{x}_i$ and z_i^n contribute to the output tensor Z^n from unstructured news data. θ denotes the parameters of the CatBoost model being optimized, while $F_{m-1}(\mathbf{x}_i)$ is the ensemble prediction from the first $m-1$ trees, and γ is for penalize.

To transform unstructured Financial report data into a structured rating representation tensor, we employ a Large Language Model (LLM) as an NLP tensor extractor. Specifically, there are two steps.

First, given an input document U, the LLM generates a five-dimensional rating vector $R \in \mathbb{R}^5$. Second, to ensure compatibility with downstream tasks, we map R into a structured feature space optimized for predictive modeling. This transformation is defined as:

$$Z^r = f_{\text{proj}}(f_{\text{rate}}(U)) \tag{4}$$

where $f_{\text{rate}}(\cdot)$ leverages contextual embeddings to infer rating scores, and Z^r represents the final structured representation used for model training and inference. Finally, we obtained four types of tensors for the next stage of encoding: tensors from historical data Z^h, macro index Z^m, news data Z^n, and financial reports Z^r.

[3] The data contains no instance where daily change in close price is exactly zero.

4.2 Tensor Encoding

The influence of a specific event in an event-driven model will usually persist for an extended period, rather than being limited to a single point. Therefore, following the approaches of [22, 27], we apply a weighted moving average (WMA) to model the decay of influence for data that do not have daily granularity. This assigns higher weights to more recent observations, allowing us to model the diminishing impact of an event over time:

$$WMA_t = \frac{\sum_{a=1}^{b} a \cdot S_{t-(b-a)}}{\sum_{a=1}^{b} a} \tag{5}$$

Here, b denotes the fixed window size; and $S_{t-(b-a)}$ represents the observation (e.g., news sentiment score) at time $t-a$. This approach allows us to model the extended influence over subsequent a days.

Once the WMA is calculated, we apply this Temporal Fusion (TF) to all tensors:

$$\mathrm{Z}_{daily}^{h,m,n,r} = \mathrm{TF}_t(Z^{h,m,n,r}) \tag{6}$$

We then concatenate the resulting tensors to form the final feature set $\mathcal{X} = \mathrm{Concat}(\mathrm{Z}_{daily}^{h}, \ldots, \mathrm{Z}_{daily}^{r}) \in \mathbb{R}^{T \times D}$.

To decode cross-modal interactions in financial tensors, we propose an interpretable feature selection framework, combining temporal sparsity and stability analysis. Given input tensor $\mathcal{X} \in \mathbb{R}^{T \times D}$, the pipeline proceeds through four stages:

Correlation-Guided Pre-selection We first eliminate multicollinear features through mean absolute correlation thresholding:

$$\Phi_{\text{corr}} = \left\{ d \in [1, D] \;\middle|\; \frac{1}{D-1} \sum_{\substack{d'=1 \\ d' \neq d}}^{D} |\rho(\mathbf{x}_d, \mathbf{x}_{d'})| < \tau_{\text{corr}} \right\} \tag{7}$$

where correlation score τ_{corr} is computed from $\mathcal{X}$'s correlation matrix.

Temporal Feature Expansion Next, we construct lagged features to capture delayed market responses through first-order temporal convolution:

$$\tilde{\mathcal{X}}_{t,d} = \begin{bmatrix} \mathcal{X}_{t',d} \\ \mathcal{X}_{t'-1,d} \end{bmatrix} \quad \forall d \in \Phi_{\text{corr}},\, t' \in \{2, \ldots, T'\} \tag{8}$$

Multi-Task Group Sparsity We then solve the convex temporal group LASSO objective [37, 46]:

$$\min_{W \in \mathbb{R}^{|\Phi_{\text{corr}}| \times N}} \underbrace{\frac{1}{2T'} \|Y - \tilde{\mathcal{X}} W\|_F^2}_{\text{Reconstruction Error}} + \alpha \underbrace{\sum_{d=1}^{|\Phi_{\text{corr}}|} \|W_d\|_2}_{\text{Cross-Target Sparsity}} \tag{9}$$

where Y is the matrix of ground truth outputs for target series. $\| \cdot \|_F$ denotes the Frobenius norm, and $\frac{1}{2T'}$ normalizes the squared error over the total number of time steps. The group LASSO penalty $\sum_d \|W_d\|_2$ encourages sparsity at the feature level across all targets, meaning that only the most relevant features are selected.

4.3 Tensor Interpretation

Stability Selection Finally, we retain features with persistent predictive power across temporal folds through majority voting:

$$\Phi_{\text{final}} = \left\{ d \in \Phi_{\text{corr}} \;\middle|\; \frac{1}{K} \sum_{k=1}^{K} \mathbb{I}\left(\|W_d^{(k)}\|_2 > 0 \right) \geq 0.8 \right\} \tag{10}$$

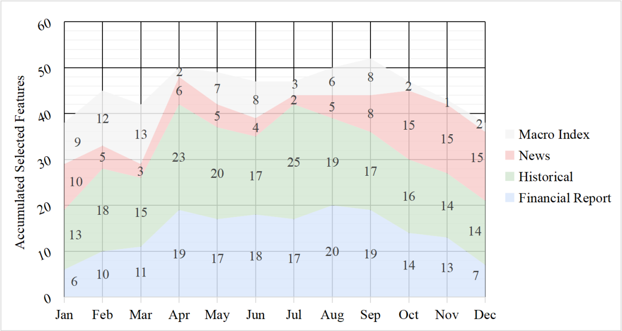

Figure 3: Demonstration of how tensor interpretation accumulates feature values over time. Here, we assume that CMTF undergoes monthly training iterations; a higher label count indicates greater importance for that period.

where K is the number for temporal splits preserve chronological order in $\mathcal{X}'$, the output tensor which will be used in the final step.

As demonstrated in Fig. 3, the relative frequency of each feature value determines its importance. As the training iterates monthly, it identifies key features that are important in specific time windows. This accumulation highlights stable or strongly correlated features, which improves the interpretation of cross-modal interactions over time.

4.4 Event-Driven Forecast Model

The event-driven forecast model includes a transformer and an optimizer for rapid hyperparameter updates using the filtered feature $\mathcal{X}'$ from the tensor interpretation.

Transformer The encoder in our transformer model consists of several key components, including multi-head attention (MHA), feed-forward layers (FFN), positional encoding (PE), and layer normalization.

Let $H_l \in \mathbb{R}^{T \times d_{\text{model}}}$ denote the input to the attention layer at encoder layer l, where d_{model} is the feature dimension of each token. This input H_l includes the original feature embedding combined with positional encodings, following standard practice in transformer architectures [35, 10].

For each attention head $h \in \{1, \ldots, H\}$, we compute:

$$Q_h = H_l W_h^Q, \quad K_h = H_l W_h^K, \quad V_h = H_l W_h^V \tag{11}$$

$$\text{Attention}_h = \text{softmax}\left(\frac{Q_h K_h^T}{\sqrt{d_k}} \right) V_h \tag{12}$$

$$\text{MHA}(H_l) = \text{Concat}(\text{Attention}_1, \ldots, \text{Attention}_H) W^O \tag{13}$$

where Q, K, and V represent the query, key, and value matrices, respectively. The attention scores are normalized using the softmax function and applied to the value matrix V to produce the output.

Each encoder layer also contains a position-wise feed-forward network. This consists of two fully connected layers with a ReLU activation function applied between them. The feed-forward network is applied independently to each position in the sequence. The operation can be formally written as:

$$\text{FFN}(x) = \text{ReLU}(W_1 x + b_1) W_2 + b_2 \tag{14}$$

where W_1 and W_2 are learnable weights, and b_1 and b_2 are bias terms. Then, the positional encoding function PE is given by:

Table 1: Data Summary.

Category	**Historical Data**	**Macro index**			**News**	**Financial Reports**
Subject	From 5 Companies	From 2 Countries			From 5 Companies	From 5 Companies
Subcategory	Historical Prices	1/10Y-Bond-Yield	GDP	CPI	Company News	Financial Reports
Detail	OHLCV	OHLC	Index	Value	Text	Text
Granularity	Daily	Daily	Quarterly	Monthly	Daily	Quarterly

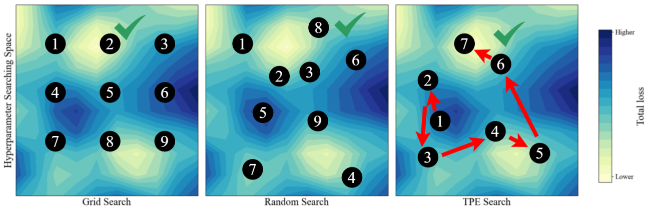

Figure 4: Hyperparameter search strategies over loss landscape: grid search, random search, and tree-structured parzen estimator (TPE).

$$\mathrm{PE}_{(pos,2\delta)} = \sin\left(\frac{pos}{10000^{2\delta/d_{\text{model}}}}\right) \tag{15}$$

$$\mathrm{PE}_{(pos,2\delta+1)} = \cos\left(\frac{pos}{10000^{2\delta/d_{\text{model}}}}\right) \tag{16}$$

where pos is the position of the token, and δ is the dimension index of the positional encoding.

Finally, the output is taken from the last time step of the sequence, and a linear layer is applied to produce the predictions:

$$\hat{P}_{t+1} = \mathrm{Linear}(x_T) \tag{17}$$

Here, $x_T \in \mathbb{R}^{d_{\text{model}}}$ refers to the hidden state from the final encoder layer at the last chronological time step, and $\hat{P}_{t+1}$ represents CMTF's prediction

Optimizer To enable rapid updates and efficient training, we use Optuna [1] as the optimization framework. Optuna utilizes an asynchronous successive halving algorithm, which is equipped with different estimators to search for the local optimum in the hyperparameter space. The pruning criterion for each trial is defined as:

$$\mathrm{Prune}(t) = \begin{cases} \text{True}, & \text{if } \frac{r_k}{r_{k-1}} > \gamma^{1/\eta} \\ \text{False}, & \text{otherwise} \end{cases} \tag{18}$$

where r_k is the trial's intermediate value at step k, η is the reduction factor, and γ is the threshold.

Here, in the CMTF framework, we apply the default estimator, Tree-structured Parzen Estimator (TPE). The TPE estimator models the probability of a set of hyperparameters x given the value of the objective function y as:

$$p(x|y) = \frac{\ell(x)}{\ell(x) + g(x)}, \quad y \sim \mathrm{Gamma}(k, \theta) \tag{19}$$

where $\ell(x)$ and $g(x)$ represent the likelihood functions for good and bad hyperparameter configurations, respectively, and y follows a Gamma distribution with shape k and scale parameter θ. As shown in Fig. 4, TPE demonstrates greater efficiency in identifying locally optimal hyperparameter combinations.

5 Empirical Analysis of CMTF

In this section, we describe our empirical analysis of CMTF. The analysis is designed to answer three questions:

RQ1 How effectively does CMTF forecast financial markets?
RQ2 How effective are the individual modules of CMTF?
RQ3 How does sensitivity to the tensor interpretation module affect performance?

5.1 Dataset

Table 1 summarizes our raw data, and Table 2 introduce the preprocessed data. The raw data covers 1360 days (02/04/2019–05/22/2024). We integrate structured financial data with unstructured textual sources from five representative UK-headquartered multinational corporations listed in the FTSE 100 index: *Shell*, *Unilever*, *British American Tobacco*, *BP*, and *Diageo*. Macro indexes are chosen from the US and UK to represent the macroeconomics of the world and the target market.

To train CMTF, the data is partitioned chronologically into distinct splits: 804 training days (02/2019 – 07/2022), 268 validation days (08/2022 – 09/2023), and 268 test days (10/2023 – 05/2024). The final tensor structure preserves cross-modal interactions between market movements (price), macro-indexes (bond/GDP/CPI), and corporate disclosures (news/reports). For more details, please refer to our GitHub.

5.2 Configuration

All experiments were performed on an Nvidia GeForce RTX 4060 laptop GPU with CUDA version 12.6. The lookback window b for the weighted moving average WMA_t is set to 30, while the temporal feature expansion window $T^{'}$ is 90. The five-dimensional rating vector comprises Risk, Market Conditions, Regulation, ESG, and Innovation, with rating scores ranging from 1 to 9, extracted from the pretrained LLM (Llama-3.1-8B). Missing values are handled using linear interpolation for numerical data, while zero-imputation is applied to text embeddings.

To optimize the Transformer model, we employ Optuna [1] for efficient hyperparameter tuning, covering both the model architecture and the training parameters. For the model architecture, the embedding dimension is selected from $\{32, 64, 128, 256, 512, 1024\}$. The number of attention heads is chosen from $\{2, 4, 8, 16\}$, ensuring divisibility by d_{model}. The model consists of multiple transformer encoder layers, with num_layers set from $\{1, 2, 4, 8\}$. The FFN layer dimension is optimized from $\{256, 512, 1024, 2048, 4096\}$.

For training, the learning rate is searched within {1e-5, 5e-5, 1e-4, 5e-4, 1e-3, 5e-3, 1e-2}. The batch size is chosen from $\{32, 64, 128\}$, balancing computational efficiency and model convergence. The number of training epochs is set from $\{10, 20, 50, 100\}$ to regulate training duration and stability. For baselines, ARIMA was configured with automatic order selection, which finds the best combination of (p, d, q) by evaluating multiple models; LSTM used sequential inputs with 50 units, ReLU activation, and 200 training epochs; and SVR used a linear kernel with separate models trained for each target variable, with an average result calculated.

Table 2: Data Statistics.

Category	**Historical Data**	**Macro index**	**News**	**Financial Reports**
Data Type	**Structured Data**		**Unstructured Data**	
# Extracted Structured Features	-		2 Labels & 1 Score	5 Types of Ratings
# Total Features	25	20	15	25
Time Span	02/04/2019 – 05/22/2024 (1360 Days)			
Data Split	0.6 : 0.2 : 0.2 (Train, Validation, Test)			
# Day Split	804 : 268 : 268 (Train, Validation, Test)			

Table 3: Classification performance comparison.

	Zero	Linear	ARIMA	RF	SVR	LSTM	CMTF
Precision (%) ↑	48.71	49.33	47.13	51.54	50.10	49.49	51.04
Recall (%) ↑	48.86	78.21	38.58	71.10	77.16	7.41	84.88
F1 Score ↑	0.49	0.61	0.42	0.60	0.61	0.13	0.64

5.3 Baseline Comparison Models

We benchmark our framework against various forecasting models that span the methodological spectrum from interpretable linear statistical models to neural network architectures. These are chosen to rigorously test our framework's ability to integrate multimodal data and temporal dynamics beyond conventional approaches.

Null Model
Zero Change: Price prediction: tomorrow's close will equal today's close: $p_{t+1} = p_t$. Direction prediction: tomorrow's direction will equal today's direction.

Classical Statistical Models
Linear Regression: Models linear relationships between dependent and independent variables by minimizing the sum of squared residuals to fit an optimal hyperplane [38].
ARIMA: Combines autoregressive (AR), differencing (I), and moving average (MA) components to capture temporal dependencies, trends, and seasonality in non-stationary time series [3].

Machine Learning Approaches
Random Forest: An ensemble method that aggregates predictions from multiple decorrelated decision trees, reducing overfitting via bootstrap aggregation and feature randomization [4].
Support Vector Regression (SVR): Extends support vector machines to regression tasks by mapping inputs to a high-dimensional space and optimizing a margin-sensitive loss function [11].

Deep Learning Architectures
LSTM: A recurrent neural network variant with gating mechanisms (input, output, forget gates) to model long-term sequential dependencies while mitigating vanishing gradients [14].
Encoder-only transformer: Adapts self-attention [35] mechanisms for time series by encoding positional information and temporal relationships, following techniques in [10].

5.4 Evaluation Metrics

Following the approaches taken in previous studies [29, 31, 23, 9, 28], we assess our result using Precision, Recall, and F1 score to evaluate model performance. For all three metrics, higher values indicate better model performance.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{20}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{21}$$

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{22}$$

In our classification scheme: true positive (TP) indicates we correctly predicted an increase in price; true negative (TN) indicates we correctly predicted a decrease in price; false positive (FP) indicates we incorrectly predicted a price increase when the price decreased; and false negative (FN) indicates we incorrectly predicted a price decrease when the price increased. Note that we focus only on classification metrics and deliberately avoid using error metrics that are often applied for regression tasks. This is because a zero change model ($p_{t+1} = p_t$) will return a low root mean squared error or mean absolute percentage error.

5.5 RQ1 Performance Comparison

Table 3 presents a comprehensive performance evaluation of our proposed CMTF against different baselines. To test the effectiveness of our framework, we do not enable the Tensor Representation module: we only include Tensor Encoding, Tensor Interpreting, and Transformer forecasting in the classification settings. In classification, our proposed CMTF framework exhibits the highest recall of 84.88% and an F1-score of 0.64, outperforming all baselines. This highlights its strength in capturing sequential dependencies and leveraging multimodal data sources to improve predictive accuracy. Random Forest achieves an F1-score of 0.60, indicating its effectiveness in feature selection and ensemble learning, though its recall 71.10% remains lower than CMTF.

The experimental results reveal a key advantage of CMTF: its ability to effectively integrate heterogeneous data sources while maintaining strong predictive capabilities. Unlike traditional models that rely on a single data modality, CMTF exploits tensor factorization to capture cross-modal dependencies, leading to superior classification performance.

5.6 RQ2 Ablation Study

To further investigate the contribution of different components in our CMTF framework, we performed an ablation study with various configurations, as shown in Table 4. Here, Macro Scaling is enabled by default. The experiments analyze the impact of three key factors: Tensor Interpretation module (*I*), news data (*N*), and financial reports (*R*). For classification, the highest recall 80.09% and best F1-score 0.61 occur when Tensor Interpretation is disabled (*-I*) but both News and Financial Reports are included (*+N, +R*). This suggests that textual modalities are crucial for predicting market movement direction, likely due to their ability to capture sentiment and fundamental shifts.

5.7 RQ3 Module Sensitivity

To understand the impact of Tensor Interpreting (*+I*), we conduct an ablation study to compare baseline methods with and without this component. From Table 5, we find that the impact of *+I* is nuanced. Precision remains relatively stable across models, indicating that

Table 4: Ablation study on CMTF with different configurations (+/-, *I*/*N*/*R*), denoting with/without tensor interpreting (I), news (N), and financial reports (R).

Metric	+ *I*				- *I*			
	+N+R	*+N-R*	*-N+R*	*-N-R*	*+N+R*	*+N-R*	*-N+R*	*-N-R*
Precision (%) ↑	51.44	49.40	49.69	49.57	49.91	50.18	51.29	49.79
Recall (%) ↑	45.51	49.40	72.01	69.46	80.09	60.93	65.42	72.16
F1 Score ↑	0.48	0.49	0.59	0.58	0.61	0.55	0.58	0.59

Table 5: Ablation study comparing base methods with Tensor Interpreting (+ *I*).

Metric	Linear Regression		ARIMA		Random Forest		SVR		LSTM		Transformer	
	Base	+I	Base	+I	Base	+I	Base	+I	Base	+I	Base	+I
Precision (%) ↑	49.55	49.22	47.13	47.13	50.41	49.84	49.61	51.62	50.40	48.24	51.29	49.69
Recall (%) ↑	57.79	61.72	38.58	38.58	75.19	69.59	76.25	67.62	19.21	45.54	65.42	72.01
F1 Score ↑	0.53	0.55	0.42	0.42	0.60	0.58	0.60	0.59	0.28	0.47	0.58	0.59

Tensor Interpreting does not compromise class separability. In contract, recall exhibits noticeable improvements, particularly for Transformer and LSTM, suggesting that *+I* aids in identifying more relevant patterns for classification. The results confirm that Tensor Interpretation enhances performance, particularly for models that rely heavily on feature transformations (e.g., SVR, LSTM, Transformer).

6 Discussion

Although we have evaluated our framework across multiple time series forecasting models, several challenges remain. One key limitation is the absence of a publicly available standardized dataset for multimodal and cross-modal financial forecasting. For example, in [21], they used data from [41], which includes only tweets and historical prices, limiting its applicability to broader multimodal scenarios. Although [19] discussed methods for handling different data granularities, it does not address the integration of unstructured data, such as textual information, into forecasting models. Another major challenge is data privacy. In a previous study [5], they compiled a diverse dataset that included financial events, news, historical prices, and knowledge graph data. However, due to privacy concerns, the dataset was not made publicly available, restricting reproducibility and benchmarking opportunities for future research.

Given these constraints, our evaluation focuses primarily on comparative baseline experiments after the feature engineering stage. Addressing these challenges, either by developing open multimodal financial datasets or by refining privacy-preserving data-sharing mechanisms, would be a crucial step forward in the future.

Future work for CMTF could explore the correlation between financial reports (R) and news (N), as their sentiment may be inherently linked. While current results show combined effects, analyzing their individual contributions may uncover deeper interactions and improve the interpretability of the CMTF framework. Additionally, the strong performance of simpler models such as SVR suggests that unstructured data may be less relevant for next-day price level prediction. This points to the need for task-specific model selection, where increased complexity is justified only when it adds meaningful predictive value. Exploring when and why simpler models outperform could provide valuable insights into the limits and optimal use cases of multi-modal approaches like CMTF.

7 Conclusion

We introduce Cross-Modal Temporal Fusion (CMTF), a transformer-based deep learning framework for financial market forecasting. To effectively capture the interactions between historical price trends, macro indexes, and textual financial data, CMTF incorporates specialized components. These include: (1) an attention-based cross-modal fusion mechanism that dynamically weighs the contribution of different modalities, (2) a tensor interpretation module to extract relevant cross-modal features, and (3) an auto-training scheme to streamline model iteration and optimization. Using real-world financial data sets, we demonstrate that CMTF outperforms all baselines on the classification task. Lastly, we explore the interpretability of our model, highlighting how CMTF can (i) analyze the relative importance of different modality data and (ii) adapt to evolving market dynamics through its feature selection mechanisms. For industrial users, CMTF is more than just a financial market forecasting model; it serves as a robust framework to efficiently handle multimodal data.

Acknowledgements

This work was supported by UK Research and Innovation (UKRI) Innovate UK grant number 10094067: Stratlib.AI - A trusted machine learning platform for asset and credit managers.

References

[1] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. Optuna: A next-generation hyperparameter optimization framework. In *25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2623–2631, 2019. doi:10.1145/3292500.3330701.

[2] N. Barberis and R. Thaler. A Survey of Behavioral Finance. In *Handbook of the Economics of Finance*, volume 1, Part B: Financial Markets and Asset Pricing, chapter 18, pages 1053–1128. Elsevier, 2003. doi:10.1016/S1574-0102(03)01027-6.

[3] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung. *Time series analysis: forecasting and control*. John Wiley and Sons, Hoboken, NJ, 2015.

[4] L. Breiman. Random forests. *Machine learning*, 45:5–32, 2001. doi:10.1023/A:1010933404324.

[5] D. Cheng, F. Yang, S. Xiang, and J. Liu. Financial time series forecasting with multi-modality graph neural network. *Pattern Recognition*, 121 (108218), 2022. doi:10.1016/j.patcog.2021.108218.

[6] J. Choi, S. Yoo, X. Zhou, and Y. Kim. Hybrid information mixing module for stock movement prediction. *IEEE Access*, 11:28781–28790, 2023. doi:10.1109/ACCESS.2023.3258695.

[7] A. Cowles 3rd. Can stock market forecasters forecast? *Econometrica*, 1(3):309–324, 1933. doi:10.2307/1907042.

[8] C. Cui, X. Li, C. Zhang, W. Guan, and M. Wang. Temporal-relational hypergraph tri-attention networks for stock trend prediction. *Pattern Recognition*, 143(109759), 2023. doi:10.1016/j.patcog.2023.109759.

[9] S. Deng, N. Zhang, W. Zhang, J. Chen, J. Z. Pan, and H. Chen. Knowledge-driven stock trend prediction and explanation via temporal convolutional network. In *World Wide Web Conference (WWW)*, pages 678–685, 2019. doi:10.1145/3308560.3317701.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 4171–4186, 2019. doi:10.18653/v1/N19-1423.

[11] H. Drucker, C. J. Burges, L. Kaufman, A. Smola, and V. Vapnik. Support vector regression machines. In *10th International Conference on Neural Information Processing Systems*, pages 155–161, 1996. URL https://proceedings.neurips.cc/paper_files/paper/1996/file/d38901788c533e8286cb6400b40b386d-Paper.pdf.

[12] E. F. Fama. The behavior of stock-market prices. *The journal of Business*, 38(1):34–105, 1965. URL http://www.jstor.org/stable/2350752.

[13] E. F. Fama. Efficient Capital Markets: A Review of Theory and Empirical Work. *The journal of Finance*, 25(2):383–417, 1970. URL https://www.jstor.org/stable/2325486.

[14] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. doi:10.1162/neco.1997.9.8.1735.

[15] X. Hou, K. Wang, C. Zhong, and Z. Wei. ST-Trader: A spatial-temporal deep neural network for modeling stock market movement. *IEEE/CAA Journal of Automatica Sinica*, 8(5):1015–1024, 2021. doi:10.1109/JAS.2021.1003976.

[16] Z. Hu, W. Liu, J. Bian, X. Liu, and T.-Y. Liu. Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction. In *ACM International Conference on Web Search and Data Mining*, pages 261–269, 2018. doi:10.1145/3159652.3159690.

[17] Z. Jin, Y. Yang, and Y. Liu. Stock closing price prediction based on sentiment analysis and LSTM. *Neural Computing and Applications*, 32:9713–9729, 2020. doi:10.1007/s00521-019-04504-2.

[18] P. Kumar. Deep reinforcement learning for high-frequency market making. In *14th Asian Conference on Machine Learning (ACML)*, pages 531–546. PMLR, 2023. URL https://proceedings.mlr.press/v189/kumar23a/kumar23a.pdf.

[19] B. Lim, S. Ö. Arık, N. Loeff, and T. Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4):1748–1764, 2021. doi:https://doi.org/10.1016/j.ijforecast.2021.03.012.

[20] H. Lin, D. Zhou, W. Liu, and J. Bian. Learning multiple stock trading patterns with temporal routing adaptor and optimal transport. In *ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1017–1026, 2021. doi:10.1145/3447548.3467358.

[21] R. C. Mandal, R. Kler, A. Tiwari, I. Keshta, M. R. Abonazel, E. M. Tageldin, and M. S. Umaralievich. Enhancing stock price prediction with deep cross-modal information fusion network. *Fluctuation and Noise Letters*, 23(2)(2440017), 2024. doi:10.1142/S0219477524400170.

[22] M. Ortu, N. Uras, C. Conversano, S. Bartolucci, and G. Destefanis. On technical trading and social media indicators for cryptocurrency price classification through deep learning. *Expert Systems with Applications*, 198(116804), 2022. doi:10.1016/j.eswa.2022.116804.

[23] Y. Pei, J. Zheng, and J. Cartlidge. Dynamic Graph Representation with Contrastive Learning for Financial Market Prediction: Integrating Temporal Evolution and Static Relations. In *Proceedings of the 17th International Conference on Agents and Artificial Intelligence (ICAART)*, volume 2, pages 298–309, Feb. 2025. doi:10.5220/0013154700003890.

[24] A. Picasso, S. Merello, Y. Ma, L. Oneto, and E. Cambria. Technical analysis and sentiment embeddings for market trend prediction. *Expert Systems with Applications*, 135:60–70, 2019. doi:10.1016/j.eswa.2019.06.014.

[25] S.-H. Poon and C. W. J. Granger. Forecasting volatility in financial markets: A review. *Journal of Economic Literature*, 41(2):478–539, 2003. URL https://www.jstor.org/stable/3216966.

[26] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin. CatBoost: unbiased boosting with categorical features. In *32nd International Conference on Neural Information Processing Systems*, pages 6639–6649, 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/14491b756b3a51daac41c24863285549-Paper.pdf.

[27] S. A. Saragih, A. Munar, and W. R. Hasibuan. Forecasting the Amount of Corn Production in North Sumatra Based on 2017–2021 Data Using The Single and Double Exponential Smoothing Method (Case Study of Central Bureau of Statistics of North Sumatra). *Journal of Artificial Intelligence and Engineering Applications*, 3(2):614–617, 2024. doi:10.59934/jaiea.v3i2.449.

[28] R. Sawhney, S. Agarwal, A. Wadhwa, and R. Shah. Deep attentive learning for stock movement prediction from social media text and company correlations. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8415–8426, 2020. doi:10.18653/v1/2020.emnlp-main.676.

[29] O. Shobayo, S. Adeyemi-Longe, O. Popoola, and B. Ogunleye. Innovative Sentiment Analysis and Prediction of Stock Price Using FinBERT, GPT-4 and Logistic Regression: A Data-Driven Approach. *Big Data and Cognitive Computing*, 8(11)(143), 2024. doi:10.3390/bdcc8110143.

[30] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *2nd International Conference on Learning Representation (ICLR) - Workshop Poster*, 2014. doi:10.48550/arXiv.1312.6034.

[31] J. Singh and M. Khushi. Feature learning for stock price prediction shows a significant role of analyst rating. *Applied System Innovation*, 4(1)(17), 2021. doi:10.3390/asi4010017.

[32] M. Tavakoli, R. Chandra, F. Tian, and C. Bravo. Multi-modal deep learning for credit rating prediction using text and numerical data streams. *Applied Soft Computing*, 171(112771), 2025. doi:10.1016/j.asoc.2025.112771.

[33] S. J. Taylor. *Modelling financial time series*. World Scientific, London, 2nd edition, 2007. doi:10.1142/6578.

[34] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit, et al. MLP-Mixer: An all-MLP architecture for vision. In *35th Conference on Neural Information Processing Systems*, pages 24261–24272, 2021. URL https://proceedings.nips.cc/paper/2021/file/cba0a4ee5ccd02fda0fe3f9a3e7b89fe-Paper.pdf.

[35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *31st Conference on Neural Information Processing Systems*, volume 30, pages 6000–6010, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[36] J. Wang, T. Sun, B. Liu, Y. Cao, and H. Zhu. CLVSA: A Convolutional LSTM Based Variational Sequence-to-Sequence Model with Attention for Predicting Trends of Financial Markets. In *28th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3705–3711, 2019. doi:10.24963/ijcai.2019/514.

[37] Y. Wang, Y. Chen, K. Jamieson, and S. S. Du. Improved active multi-task representation learning via Lasso. In *40th International Conference on Machine Learning (ICML)*, pages 35548–35578, 2023. URL https://proceedings.mlr.press/v202/wang23b/wang23b.pdf.

[38] S. Weisberg. *Applied linear regression*, volume 528 of *Wiley Series in Probability and Statistics*. John Wiley and Sons, Hoboken, NJ, 2005. doi:10.1002/0471704091.

[39] K. Wood, S. Giegerich, S. Roberts, and S. Zohren. Trading with the momentum transformer: An intelligent and interpretable architecture. Preprint arXiv:2112.08534, 2021.

[40] H. Wu, J. Xu, J. Wang, and M. Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *35th International Conference on Neural Information Processing Systems*, pages 22419–22430, 2021. URL https://proceedings.neurips.cc/paper/2021/file/bcc0d400288793e8bdcd7c19a8ac0c2b-Paper.pdf.

[41] Y. Xu and S. B. Cohen. Stock movement prediction from tweets and historical prices. In *56th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 1970–1979, 2018. doi:10.18653/v1/P18-1183.

[42] Z. You, P. Zhang, J. Zheng, and J. Cartlidge. Multi-relational graph diffusion neural network with parallel retention for stock trends classification. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6545–6549, 2024. doi:10.1109/ICASSP48485.2024.10447394.

[43] D. Zhou, L. Zheng, Y. Zhu, J. Li, and J. He. Domain adaptive multi-modality neural attention network for financial forecasting. In *The Web Conference*, pages 2230–2240, 2020. doi:10.1145/3366423.3380288.

[44] F. Zhou, H.-m. Zhou, Z. Yang, and L. Yang. EMD2FNN: A strategy combining empirical mode decomposition and factorization machine based neural network for stock market trend prediction. *Expert Systems with Applications*, 115:136–151, 2019. doi:10.1016/j.eswa.2018.07.065.

[45] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *35th AAAI Conference on Artificial Intelligence*, volume 35(12), pages 11106–11115, 2021. doi:10.1609/AAAI.V35I12.17325.

[46] Y. Zhou, R. Jin, and S. C.-H. Hoi. Exclusive Lasso for multi-task feature selection. In *13th International Conference on Artificial Intelligence and Statistics*, pages 988–995. PMLR, 2010.

[47] Y. Zu, J. Mi, L. Song, S. Lu, and J. He. Finformer: A Static-dynamic Spatiotemporal Framework for Stock Trend Prediction. In *IEEE International Conference on Big Data (BigData)*, pages 1460–1469. IEEE, 2023. doi:10.1109/BigData59044.2023.10386751.